#2

Docket No. 826.1721

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:

Hironori YAHAGI

Serial No.: To be assigned

Filed: March 27, 2001

For: APPARATUS CONVERTING A STRUCTURED DOCUMENT HAVING A HIERARCHY

Group Art Unit: Unassigned

Examiner: Unassigned

SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN APPLICATION IN ACCORDANCE WITH THE REQUIREMENTS OF 37 C.F.R. §1.55

Assistant Commissioner for Patents
Washington, D.C. 2023l

Sir:

In accordance with the provisions of 37 C.F.R. §1.55, the applicant submits herewith a certified copy of the following foreign application:

Japanese Patent Application No. 2000-296161
Filed: September 28, 2000.

It is respectfully requested that the applicant be given the benefit of the foreign filing date as evidenced by the certified papers attached hereto, in accordance with the requirements of 35 U.S.C. §119.

Respectfully submitted,

STAAS & HALSEY LLP

Date: March 27, 2001

By: _____

James D. Halsey, Jr.
Registration No. 22,729

700 Eleventh Street, N.W.
Suite 500
Washington, D.C. 20001
(202) 434-1500

PATENT OFFICE

JAPANESE GOVERNMENT

This is to certify that the annexed is a true copy of the following application as filed with this Office.

Date of Application: September 28, 2000

Application Number: Patent Application No. 2000-296161

Applicant(s): FUJITSU LIMITED

December 22, 2000

Commissioner,

Patent Office Kozo OIKAWA

Certificate No. 2000-3105882

日　本　国　特　許　庁
PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: ２０００年　９月２８日

出願番号
Application Number: 特願２０００－２９６１６１

出願人
Applicant (s): 富士通株式会社

２０００年１２月２２日

特許庁長官
Commissioner,
Patent Office 及川耕造

出証番号　出証特２０００－３１０５８８２

【書類名】　特許願
【整理番号】　0051425
【提出日】　平成12年 9月28日
【あて先】　特許庁長官殿
【国際特許分類】　G06F 12/00
G06F 17/30
【発明の名称】　階層構造の構造化文書を変換する装置
【請求項の数】　5
【発明者】
【住所又は居所】　神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内
【氏名】　矢作　裕紀
【特許出願人】
【識別番号】　000005223
【氏名又は名称】　富士通株式会社
【代理人】
【識別番号】　100074099
【住所又は居所】　東京都千代田区二番町８番地２０　二番町ビル３Ｆ
【弁理士】
【氏名又は名称】　大菅　義之
【電話番号】　03-3238-0031
【選任した代理人】
【識別番号】　100067987
【住所又は居所】　神奈川県横浜市鶴見区北寺尾７－２５－２８－５０３
【弁理士】
【氏名又は名称】　久木元　彰
【電話番号】　045-573-3683
【手数料の表示】
【予納台帳番号】　012542

【納付金額】　　21,000円

【提出物件の目録】

【物件名】　　明細書　1

【物件名】　　図面　1

【物件名】　　要約書　1

【包括委任状番号】　9705047

【プルーフの要否】　　要

【書類名】　明細書

【発明の名称】　階層構造の構造化文書を変換する装置

【特許請求の範囲】

【請求項１】　階層構造の要素の集合で記述され、それぞれが１つ以上の要素を含む複数のレコードから成る構造化文書の情報を入力する文書入力手段と、

前記構造化文書の２つ以上のレコード間で、相対的に同じ位置にある要素の内容を接合して、新しい要素を生成する接合手段と、

前記新しい要素を含み、前記２つ以上のレコードにおける要素の相対的位置関係を継承した新しいレコードを生成する生成手段と、

前記２つ以上のレコードを前記新しいレコードに置き換えて、前記構造化文書を変換する変換手段と、

変換後の構造化文書を出力する文書出力手段と

を備えることを特徴とする変換装置。

【請求項２】　検索キーを入力するキー入力手段と、前記変換後の構造化文書を該検索キーで検索し、あるレコードの要素の内容から前記検索キーに対応する文字列が検出されたとき、該あるレコードにおける他の要素の内容から、検出された文字列の位置に対応する文字列を抽出し、該検出された文字列と抽出された文字列から該検索キーを含む変換前のレコードを復元し、検索結果として出力する検索手段とをさらに備えることを特徴とする請求項１記載の変換装置。

【請求項３】　階層構造の要素の集合で記述された構造化文書の情報を入力する文書入力手段と、

前記構造化文書の情報を格納する格納手段と、

前記構造化文書において、ある要素の１段下の層で連続して並ぶ同じ要素名の要素同士の組み合わせと、該組み合わせの各要素より下位のある層の同じ要素名の要素同士であって、該組み合わせの各要素から該ある層に至る経路上の各層において互いに同じ要素名の要素を経由するような、該ある層の要素同士の組み合わせとに含まれる各要素の内容を合成対象として接合し、複数の新しい要素を生成する接合手段と、

前記複数の新しい要素を含み、該複数の新しい要素の間で元の要素の相対的位

置関係を継承した合成部分構造を生成する生成手段と、

接合されなかった要素より上位の要素から生成された合成部分構造に含まれる新しい要素の下位に、該接合されなかった要素の複製を生成する複製手段と、

不要な元の要素を削除する削除手段と、

前記接合手段、生成手段、複製手段、および削除手段を用いて、前記構造化文書を合成部分構造から成る合成型構造化文書に変換する変換手段と、

前記合成型構造化文書を出力する文書出力手段と

を備えることを特徴とする変換装置。

【請求項４】　前記生成手段は、前記ある層に至る経路上の２つ以上の層において、連続して並ぶ同じ要素名の要素同士の組み合わせが見られないとき、前記合成部分構造を生成することを特徴とする請求項３記載の変換装置。

【請求項５】　コンピュータのためのプログラムを記録した記録媒体であって、該プログラムは、

階層構造の要素の集合で記述され、それぞれが１つ以上の要素を含む複数のレコードから成る構造化文書の２つ以上のレコード間で、相対的に同じ位置にある要素の内容を接合して、新しい要素を生成し、

前記新しい要素を含み、前記２つ以上のレコードにおける要素の相対的位置関係を継承した新しいレコードを生成し、

前記２つ以上のレコードを前記新しいレコードに置き換えて、前記構造化文書を変換する

処理を前記コンピュータに実行させることを特徴とするコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、階層構造を持つ要素の集合で記述される構造化文書の検索処理に係り、構造化文書の要素を検索するためにその文書の構造を変換する変換装置に関する。

【０００２】

【従来の技術】

構造化文書の記述形式の代表例としては、大規模データベース向けのＳＧＭＬ（Standard General Markup Language）、ＷＷＷ（World Wide Web）向けに簡便な構成を持つＨＴＭＬ（Hyper Text Markup Language）、ＳＧＭＬをインターネット向けに簡略化したＸＭＬ（eXtensible Markup Language）等がある。ＨＴＭＬは、ＷＷＷのコンテンツ形式として世界的に普及している。ＸＭＬは、ＨＴＭＬを補うものとして、最近、特に注目を浴びており、インターネット上で文書を記述するだけでなく、携帯電話、カーナビゲーション等あらゆる情報機器が交信するための媒介となりつつある。

【０００３】

ＸＭＬで記述されたＸＭＬ文書の概要については、例えば、「標準ＸＭＬ完全解説」（技術評論社，１９９８年，ｐｐ．２７－５１）に紹介されている。ＸＭＬ文書は、図２０に示すように、大きく分けて、ＸＭＬ宣言１１、文書型定義（Document Type Definition，ＤＴＤ）１２、およびＸＭＬ実現値（インスタンス）１３の３つの部分から成る。このうち、ＸＭＬ実現値の部分は、階層構造を持つ要素の集合で記述され、これらの要素を識別するマークとして、タグが用いられる。

【０００４】

図２１は、１つの要素を表すタグの書き方を示している。図２１において、要素名を含む開始タグ２１と終了タグ２２の間に記述された“要素の内容です。”の部分が要素の内容を表し、空要素タグ２３は、内容のない要素のタグを表す。また、要素の内容として平文と下位の要素が混在するような階層構造を表すタグの書き方は、図２２のようになる。図２２においては、要素ａの内容１と内容２の間に要素ｂが挿入されており、要素ａの下位に要素ｂが存在する。この場合、要素ａと要素ｂは親子関係にある。

【０００５】

さらに、要素に属性が与えられている場合は、以下に示すように、その要素の開始タグに属性名と属性値が記述される。

＜要素名　属性名１＝“属性値１”　属性名２＝“属性値２”・・・＞

また、ＸＭＬ文書は、処理上の観点から、整形式（well-formed ）と検証済み（valid ）の２つの種類に分けられる。このような２種類のＸＭＬ文書を含む構造化文書の構成と処理上の区分との関係は、図２３のようになる。図２３では、整形式ＸＭＬ文書、検証済みＸＭＬ文書、ＳＧＭＬ文書、およびＨＴＭＬ文書のそれぞれについて、宣言、文書型定義、および実現値が必須である（○）か否（△）かが示されている。例えば、整形式ＸＭＬ文書の場合は、実現値のみが必須であり、宣言と文書型定義はなくてもよい。

【０００６】

ＸＭＬ文書を解析して、ブラウザ等の他の応用ソフトウェアに渡す媒介となる役割を果たすソフトウェアは、ＸＭＬプロセッサ（ＸＭＬパーサー）と呼ばれる。ＸＭＬプロセッサの概要については、例えば、「ＯｐｅｎＤｅｓｉｇｎ２月号」（ＣＱ出版，２０００年２月，ｐｐ．３９－８５）に紹介されている。

【０００７】

図２４は、ＸＭＬプロセッサが行う処理の例を示している。図２４において、ＸＭＬプロセッサ３２は、与えられたＸＭＬ文書３１をチェックして、木構造で表されたＸＭＬ文書３３を応用ソフトウェア３４に渡す。このとき、ＸＭＬ文書３１に文書型定義が含まれていなければ、ＸＭＬ実現値のタグ付け形式のみがチェックされる。

【０００８】

このようなＸＭＬプロセッサにおいて、Ｊａｖａ（商標）言語でＸＭＬ文書を操作するためのＡＰＩ（Application Programming Interface ）には、ＳＡＸ（Simple API for XML）とＤＯＭ（Document Object Model ）の２種類がある。ＳＡＸは、ＸＭＬ文書を読みながら、文書や要素の開始や終了、文字列の出現といった事象（event ）を応用ソフトウェアに通知する事象駆動型のＡＰＩである。

【０００９】

これに対して、ＤＯＭは、汎用的なＸＭＬ操作ＡＰＩであり、ＸＭＬ文書をＤＯＭオブジェクトの木構造として、メモリ上に展開する。そして、応用ソフトウ

ェアは、このＤＯＭオブジェクトに対する操作を行うことで、ＸＭＬ文書へのアクセスを行うことができる。また、ＤＯＭオブジェクトから元のＸＭＬ文書を生成することもできる。

【００１０】

例えば、図２５のようなＸＭＬ文書からは、図２６のようなＤＯＭの木構造が生成される。図２６において、矢印は、各ノードを呼び出すためのメソッド（関数）を表し、Ｄｏｃｕｍｅｎｔ４１は、ＸＭＬ文書の全体を表現するインタフェースに対応する。

【００１１】

また、ＮｏｄｅＬｉｓｔ４２は、あるノードに属する下位の要素や文字データをＸＭＬ文書内での出現順に管理するために使用され、Ｅｌｅｍｅｎｔ４３やＴｅｘｔ４４等のインスタンスを下位のノードとして持つ。ＮａｍｅｄＮｏｄｅＭａｐ４５は、並び順に意味はないが、名前をキーにして値を参照する必要があるようなノードを収容するためのコレクションであり、ここには属性（Ａｔｔｒ４６）等が記述される。

【００１２】

ＸＭＬ文書の代表的な応用例として、データベースとしてのＸＭＬ文書のタグ検索がある。この処理では、ＸＭＬ文書で与えられた検索キーに対応する箇所が検索され、検索結果が出力される。

【００１３】

図２７は、ＤＯＭを用いた場合のタグ検索処理のフローチャートである。処理プログラムは、まず、ＸＭＬ文書を入力し（ステップＳ１）、検索キーを入力する（ステップＳ２）。次に、ＸＭＬプロセッサのインスタンスを生成し（ステップＳ３）、それを実行する（ステップＳ４）。これにより、ＸＭＬ文書のタグ構造が解析され、ＤＯＭの木構造が構築される。

【００１４】

次に、木構造をルート（根）から辿り、検索キーに対応する個所を検出して、木構造の不要な部分を削除する（ステップＳ５）。これにより、木構造のノードが削減されて、部分木が生成される。そして、得られた部分木を検索結果として

出力し（ステップＳ６）、処理を終了する。

【００１５】

大規模データベースをＸＭＬで構築した場合、図２７のタグ検索は、比較的高速に検索できる点で有効な方法である。例えば、住民票データのデータベースにおいて、出身県を検索キーとして入力し、ＤＯＭの木構造を探索して、該当する個人データの部分木を残して出力するような処理が可能になる。

【００１６】

また、図２８は、ＳＡＸを用いた場合のタグ検索処理のフローチャートである。処理プログラムは、まず、ＸＭＬ文書を入力し（ステップＳ１１）、検索キーを入力する（ステップＳ１２）。次に、ハンドラーのインスタンスを生成し（ステップＳ１３）、ＸＭＬプロセッサのインスタンスを生成して（ステップＳ１４）、ＸＭＬプロセッサを実行する（ステップＳ１５）。

【００１７】

ＸＭＬプロセッサは、ＸＭＬ文書のタグ構造を解析し、タグを検出する度にハンドラーを実行して、検索キーに対応する個所を検出する（ステップＳ１６）。そして、得られた検索結果を出力して（ステップＳ１７）、処理を終了する。

【００１８】

【発明が解決しようとする課題】

しかしながら、上述した従来のＤＯＭを用いたタグ検索には、以下のような問題がある。

【００１９】

ＤＯＭの木構造の規模が大きくなると、木構造を辿って、各要素の内容から検索キーと同じ文字列を検出するために、多大な処理時間を要する。また、ＤＯＭでは、各項目に長い文字列が出現することを予期して、長い固定長メモリ領域を確保してデータを書き込むので、木構造が大きくなると、大きな動作メモリ量が必要となる。

【００２０】

本発明の課題は、ＸＭＬ文書のような構造化文書を変換することにより、タグ検索の処理速度を向上させ、必要な動作メモリ量を削減する変換装置を提供する

ことである。

【００２１】

【課題を解決するための手段】

図１は、本発明の変換装置の原理図である。本発明の第１の局面において、変換装置は、文書入力手段５１、接合手段５２、生成手段５３、変換手段５４、および文書出力手段５５を備える。

【００２２】

文書入力手段５１は、階層構造の要素の集合で記述され、それぞれが１つ以上の要素を含む複数のレコードから成る構造化文書の情報を入力する。接合手段５２は、構造化文書の２つ以上のレコード間で、相対的に同じ位置にある要素の内容を接合して、新しい要素を生成する。

【００２３】

生成手段５３は、生成された新しい要素を含み、上記２つ以上のレコードにおける要素の相対的位置関係を継承した新しいレコードを生成する。変換手段５４は、それらの２つ以上のレコードを新しいレコードに置き換えて、構造化文書を変換する。そして、文書出力手段５５は、変換後の構造化文書を出力する。

【００２４】

レコードは、構造化文書を構成するために繰り返される単位データに対応し、構造化文書は、複数のレコードから成る。文書入力手段５１により入力された構造化文書は、接合手段５２に渡される。そして、接合手段５２は、２つ以上のレコードの同じ位置にある要素を合成対象としてそれらの内容を接合し、接合された内容を持つ新しい要素を生成して、生成手段５３に渡す。

【００２５】

次に、生成手段５３は、受け取った新しい要素を用いて、接合前の元のレコードにおける要素の相対的位置関係を継承した新しいレコードを生成し、変換手段５４に渡す。変換手段５４は、元のレコードを新しいレコードに置き換えて、変換後の構造化文書を生成し、文書出力手段５５に渡す。そして、文書出力手段５５は、受け取った構造化文書を変換結果として出力する。

【００２６】

また、本発明の第２の局面において、変換装置は、文書入力手段５１、接合手段５２、生成手段５３、変換手段５４、文書出力手段５５、格納手段５６、複製手段５７、および削除手段５８を備える。

【００２７】

文書入力手段５１は、階層構造の要素の集合で記述された構造化文書の情報を入力し、格納手段５６は、構造化文書の情報を格納する。接合手段５２は、構造化文書において、ある要素の１段下の層で連続して並ぶ同じ要素名の要素同士の組み合わせと、その組み合わせの各要素より下位のある層の同じ要素名の要素同士であって、その組み合わせの各要素からある層に至る経路上の各層において互いに同じ要素名の要素を経由するような、ある層の要素同士の組み合わせとに含まれる各要素の内容を合成対象として接合し、複数の新しい要素を生成する。

【００２８】

生成手段５３は、生成された複数の新しい要素を含み、それらの新しい要素の間で元の要素の相対的位置関係を継承した合成部分構造を生成する。複製手段５７は、接合されなかった要素より上位の要素から生成された合成部分構造に含まれる新しい要素の下位に、接合されなかった要素の複製を生成する。削除手段５８は、不要な元の要素を削除する。

【００２９】

変換手段５４は、接合手段５２、生成手段５３、複製手段５７、および削除手段５８を用いて、構造化文書を合成部分構造から成る合成型構造化文書に変換する。そして、文書出力手段５５は、合成型構造化文書を出力する。

【００３０】

文書入力手段５１により入力された構造化文書は、格納手段５６に格納される。次に、接合手段５２は、格納手段５６から構造化文書の情報を取り出し、合成対象の要素を選択する。ここでは、ある要素の１段下の層に並んでいる兄弟要素のうち、連続して並ぶ複数の同名の要素の組み合わせが第１の合成対象として選択される。また、それらの要素より下位の任意の層に複数の互いに同名の要素が存在し、その層に至る経路上においても互いに同名の要素が連なっているとき、その任意の層の同名の要素の組み合わせが第２の合成対象として選択される。

【００３１】

次に、接合手段５２は、各合成対象の組み合わせの中で同名の要素の内容を接合し、接合された内容を持つ新しい要素を生成して、生成手段５３に渡す。生成手段５３は、受け取った新しい要素を含み、それらの要素の間で元の要素の相対的位置関係を継承した合成部分構造を生成して、複製手段５７に渡す。

【００３２】

複製手段５７は、接合されなかった要素が存在する場合、その要素より上位の要素から生成された合成部分構造に含まれる、新しい要素の下位に、その要素の複製を追加する。また、削除手段５８は、接合された元の要素と複製された元の要素を削除する。変換手段５４は、接合手段５２、生成手段５３、複製手段５７、および削除手段５８を制御することにより、元の構造化文書を合成型構造化文書に変換し、文書出力手段５５に渡す。そして、文書出力手段５５は、受け取った合成型構造化文書を変換結果として出力する。

【００３３】

このような変換装置によれば、構造化文書の複数の要素が合成されて１つになるため、文書情報が圧縮され、文書を格納するためのメモリ量が削減される。また、要素の数が減ることで木構造のノードが減少するので、タグ検索の処理速度が向上する。

【００３４】

また、変換後の構造化文書においても、元の要素の相対的位置関係が継承されるため、元の階層構造を把握することができ、既存のブラウザ、ビューア等の応用ソフトウェアを適用して、従来の機能をそのまま実行することができる。言い換えれば、既存の応用ソフトウェアから見て、元の文書を変換したことが分からないような透過性が実現される。

【００３５】

例えば、図１の文書入力手段５１は、後述する図１８の入力装置８３またはネットワーク接続装置８７に対応し、図１の格納手段５６は、図１８のメモリ８２または外部記憶装置８５に対応する。また、例えば、図１８の接合手段５２、生成手段５３、変換手段５４、文書出力手段５５、複製手段５７、および削除手段

５８は、図１８のＣＰＵ（中央処理装置）８１およびメモリ８２に対応する。

【００３６】

【発明の実施の形態】

以下、図面を参照しながら、本発明の実施の形態を詳細に説明する。

まず、図２のＸＭＬ文書を処理対象として、合成の対象を指定する処理について説明する。一般に、ＸＭＬ文書は、複数のレコードの繰り返しにより構成される。例えば、図２のＸＭＬ文書では、＜ｅｖｅｎｔ＞から次の＜／ｅｖｅｎｔ＞までの部分が１つのレコードに対応する。図２のＸＭＬ文書をＸＭＬプロセッサで解析すると、図３のようなＤＯＭの木構造が得られる。ここでは、図２の８個のＥｖｅｎｔ要素のうち、最初の４個のみが明示的に示されている。

【００３７】

木構造において、一般に、合成の対象となるのは、同じ名前を持つ兄弟の要素からそれぞれ派生した部分木である。図３の木構造を見ると、Ｅｖｅｎｔｌｉｓｔ要素６１の下には、Ｅｖｅｎｔ要素６２が４個連続して存在する。各Ｅｖｅｎｔ要素６２に連なる部分木は、１つのレコードに対応する。このうち、左から２番目のＥｖｅｎｔ要素６２の下には、Ｉｎｆｏ要素６３が２個連続して存在し、１番目、３番目、および４番目のＥｖｅｎｔ要素６２では、Ｉｎｆｏ要素６３はそれぞれ１個しか存在しない。

【００３８】

２番目のＥｖｅｎｔ要素６２に連なる部分木の中では、子であるＩｎｆｏ要素６３が２つ存在し、それぞれのＩｎｆｏ要素６３が部分木を成している。このようなＥｖｅｎｔ要素６２を合成の対象にすると、合成の候補となる部分木が重なりあい、処理が煩雑になる。

【００３９】

そこで、このＥｖｅｎｔ要素６２の部分木のように、２つ以上の階層にまたがって同名の兄弟要素が連続して並ぶような場合は、合成の対象から外すことにする。したがって、連続して並ぶ同名の兄弟要素のうちの１つからある層に至る経路上の２つ以上の層において、連続して並ぶ同名の兄弟要素の組み合わせが見られない場合に、それらの兄弟要素に連なる部分木が合成の対象として指定される

。

【００４０】

実際の処理では、変換装置は、各要素のＮｏｄｅＬｉｓｔを見て、兄弟のノードの中で、同じ要素名を持つ要素が連続して出現するようなものを検出し、合成の対象から外す。図３では、Ｅｖｅｎｔ要素６２が４つ連続しているが、２番目のＥｖｅｎｔ要素６２は、下の層に２つの連続するＩｎｆｏ要素６３を持っているので、対象外になる。すると、連続したＥｖｅｎｔ要素６２は３番目および４番目のみとなり、これらにそれぞれ連なる部分木が合成の対象として指定される。

【００４１】

このように、合成の対象となる部分木が複数個得られた場合、これらの部分木は幾つかの群（グループ）に分割される。２個の部分木を１つの群にまとめた場合、図３では、１番目および２番目のＥｖｅｎｔ要素６２の部分木から成る部分Ｐ１が合成の対象外となり、３番目および４番目のＥｖｅｎｔ要素６２の部分木から成る部分Ｐ２が合成の対象となる。同様にして、図２の他のＥｖｅｎｔ要素のうち、Ｐ３およびＰ４の部分がそれぞれ合成の対象となる。

【００４２】

また、各部分木の親の要素（ここでは、Ｅｖｅｎｔ要素）の間には、親の要素の内容である文字データ（Ｔｅｘｔ）が挿入されている。これらの文字データについても、合成の対象となったＥｖｅｎｔ要素６２に付随するものを２つずつ群に分けて、接合の対象とする。

【００４３】

次に、図４のＸＭＬ文書を処理対象として、合成対象として指定された要素を合成する処理について説明する。ここでは、部分木を４つずつ群にまとめて、各群の中の要素を合成するものとする。図４のＸＭＬ文書をＸＭＬプロセッサで解析すると、図５のようなＤＯＭの木構造が得られる。

【００４４】

図５の木構造において、変換装置は、互いの先祖が同じ要素名を持ち、それ自身も同じ要素名を持つようなノード同士で、要素内容を接合して登録する。ここ

では、３つのＳｔａｒｔ要素７３が同じ要素名のＥｖｅｎｔ要素７２を親として持っているので、それらの要素内容である“８：４０”、“９：００”、および“９：３０”が接合される。

【００４５】

同様に、４つのＩｎｆｏ要素７４がＥｖｅｎｔ要素７２を親として持っているので、それらの要素内容である“出社”、“退社”、“接客”、および“会議”も接合される。このとき、各Ｅｖｅｎｔ要素７２に付随するＴｅｘｔ（親であるＥｖｅｎｔｌｉｓｔ要素７１の内容）も接合されて登録される。

【００４６】

接合される２つの要素内容の間には、境界を示すために境界文字（デリミッタ）を挿入する。一方の部分木には存在し、他方の部分木には存在しない（つまり、欠損している）要素については、接合の際に、存在している要素内容の後に境界文字だけを追加して、要素の欠損を表すことにする。この場合、新しい要素の内容には、境界文字が連続して挿入されることになる。

【００４７】

こうして、図５の部分木を合成した後の木構造は、図６のようになる。図６において、Ｅｖｅｎｔ要素７２に連なる部分木が合成部分木に対応し、Ｔｅｘｔ７５が合成部分木の要素内容を表し、“@”が境界文字に対応する。図５では、左から２番目のＥｖｅｎｔ要素７２にＳｔａｒｔ要素７３が欠損しているので、図６のＳｔａｒｔ要素７３の要素内容“８：４０@@９：００@９：３０”において、２つの@が連続して挿入されている。

【００４８】

また、図６の木構造に対応する合成後のＸＭＬ文書は、図７のようになる。図７のＸＭＬ文書では、＜ｅｖｅｎｔ＞から次の＜／ｅｖｅｎｔ＞までの部分が合成部分木のレコードに対応する。このレコードは、要素を合成して得られた合成部分構造を表す。

【００４９】

図６および図７に示されるように、合成部分木においては、要素内容が出現順に接合され、要素内容の間に境界文字@が挿入される。また、接合すべき内容が

ない場合は、空白を入れずに次の＠が挿入される。このような接合方法によれば、境界文字＠の位置と個数から元の部分木の構造を復元することができる。例えば、Ｔｅｘｔ７５において“Ａ＠Ｂ”と記述されていれば、初めの２つのＥｖｅｎｔ要素７２に対応する要素内容が“Ａ”および“Ｂ”であることが分かる。

【００５０】

図６では、Ｅｖｅｎｔ要素７２の内容“第一＠第二＠第三＠第四”は、１番目、２番目、３番目、および４番目のＥｖｅｎｔ要素７２の内容が、それぞれ、“第一”、“第二”、“第三”、および“第四”であることを表す。また、Ｉｎｆｏ要素７４の内容“出社＠退社＠接客＠会議”は、１番目、２番目、３番目、および４番目のＥｖｅｎｔ要素７２に連なるＩｎｆｏ要素７４の内容が、それぞれ、“出社”、“退社”、“接客”、および“会議”であることを表す。

【００５１】

一方、Ｓｔａｒｔ要素７３の内容“８：４０＠＠９：００＠９：３０”は、１番目、３番目、および４番目のＥｖｅｎｔ要素７２に連なるＳｔａｒｔ要素７３の内容が、それぞれ、“８：４０”、“９：００”、および“９：３０”であり、２番目のＥｖｅｎｔ要素７２にはＳｔａｒｔ要素７３が欠損していることを表す。したがって、このような合成部分木の要素内容から、図５の木構造を容易に復元することができる。

【００５２】

また、合成部分木のデータを保存する場合、変換装置は、その部分木の親から任意の要素に至るまでに経由する要素名の組み合わせと、接合した要素内容の文字列を、テーブルに登録する。このような保存方法によれば、異なる部分木に属する要素でも、経由する要素名の組み合わせが等しければ、テーブルの同じ欄にそれらの要素内容を登録することができ、接合が可能になる。

【００５３】

図８は、図６の合成部分木を登録したハッシュ表の例を示している。図８のハッシュ表では、ハッシュ値Ｈ１の欄に、Ｉｎｆｏ要素の接合された要素内容が、親の要素名Ｅｖｅｎｔと要素名Ｉｎｆｏの組み合わせとともに登録されている。また、ハッシュ値Ｈ２の欄には、Ｓｔａｒｔ要素の接合された要素内容が、要素

名ＥｖｅｎｔとＳｔａｒｔの組み合わせとともに登録されている。

【００５４】

例えば、ハッシュ値Ｈ１およびＨ２は、それぞれ、対応する要素名の組み合わせに基づいて算出される。また、こうして生成されたハッシュ表を元にして、ＤＯＭの木構造に合成部分木が追加される。

【００５５】

上述した合成処理では、ＸＭＬ文書の構造が一般的な木構造である場合を想定しているが、完全に同じ構造の部分木の繰返しにより木構造が構成されている場合は、より簡便な処理を採用することができる。この場合、ＸＭＬ文書をＤＯＭの木構造に変換して要素同士の位置関係を解析しなくても、あらかじめ要素の位置関係をソフトウェアに記述しておけば、内容を接合することができる。そこで、変換装置は、記述された位置関係からハッシュ値を計算して、ハッシュ表の該当個所に要素内容を登録する。

【００５６】

図９は、このような簡易型合成処理で用いられるハッシュ表の例を示している。簡易型合成処理は、同じ部分構造が規則的に繰り返される文書に対して適用されるため、要素の欠損がないことを前提としている。図９のハッシュ表では、ハッシュ値Ｈ３の欄に、図８のＩｎｆｏ要素と同様の要素内容が登録されているが、ハッシュ値Ｈ４の欄には、図８のＳｔａｒｔ要素とは異なり、欠損のない要素内容が登録されている。例えば、ハッシュ値Ｈ３およびＨ４は、それぞれ、対応する要素の位置または要素名等に基づいて算出される。

【００５７】

次に、合成後のＸＭＬ文書を用いたタグ検索の例について説明する。合成後のＸＭＬ文書のタグ検索において、変換装置は、文書内の要素の内容に含まれる２つの境界文字の間の文字列と検索キーの文字列とを照合し、検索キーに対応する文字列を検出する。次に、検出された文字列の前にある境界文字の順位を求め、同じ合成部分木における他の要素の内容において、その順位に対応する境界文字と次の境界文字の間の文字列を抽出する。そして、これらの文字列から合成前のＸＭＬ文書の対応する部分を復元し、検索結果として出力する。

【００５８】

図１０に示すＸＭＬ文書を処理対象とした場合、変換装置は、同じ要素名“個人”の要素を上位に持ち、かつ、それ自身が同じ要素名を持つような要素に属する内容同士を、境界文字を用いて接合する。これにより、“姓”、“名”、および“旧姓”の要素の内容が接合される。そして、元の要素のノードを消去すると、図１１のようなＸＭＬ文書が生成される。

【００５９】

次に、生成されたＸＭＬ文書のタグ検索において、ユーザが“鈴木”を検索キーとして指定すると、変換装置は、要素内容に“鈴木”を含む群の合成部分木を検索する。図１１の文書では、“鈴木”を含む群が１つだけしか存在しないので、この文書の全体が検索結果として得られる。次に、変換装置は、得られた結果において、“鈴木”に対応する部分だけを残して、他の部分は削除する。その結果、“佐藤”に対応する部分が削除され、図１２のような検索結果が出力される。

【００６０】

このような変換処理によれば、ＸＭＬ文書の複数の要素が合成されて１つになるため、文書情報が大幅に圧縮される。また、要素の数が減ることで木構造のノードが減少するので、タグ検索に要する探索時間が大幅に削減される。

【００６１】

次に、図１３から図１７までを参照しながら、変換装置が行う処理についてより詳細に説明する。

図１３は、ＸＭＬ文書の変換を含むタグ検索処理のフローチャートである。まず、ユーザは、１つの群にまとめられるレコードの数を指定する数値ｎを変換装置に入力する（ステップＳ２１）。この数値ｎは、要素の合成処理において、レコードを組み合わせて群にまとめる単位として用いられる。次に、ユーザは、処理対象となるＸＭＬ文書を入力する（ステップＳ２２）。

【００６２】

次に、変換装置は、合成対象指定処理を行って、入力されたＸＭＬ文書の中で、合成するｎ個のレコードの組み合わせ（群）を指定し（ステップＳ２３）、要

素の合成処理を行う（ステップＳ２４～Ｓ２７）。

【００６３】

合成処理において、変換装置は、まず、指定されたすべての組み合わせについての合成が終了したか否かをチェックする（ステップＳ２４）。合成されていない組み合わせがあれば、その組み合わせに含まれるｎ個のレコードの間で、相対的に同じ位置関係にある要素の内容を接合し、１つ以上の新しい要素を生成する（ステップＳ２５）。

【００６４】

次に、生成された新しい要素を元に、元のレコードと同じような要素の相対的位置関係を継承した新しいレコードを生成する（ステップＳ２６）。そして、元のｎ個のレコードを削除して、新しいレコードに置き換え（ステップＳ２７）、ステップＳ２４以降の処理を繰り返す。

【００６５】

ステップＳ２４において、すべての組み合わせについての合成が終了すると、復元処理を行って（ステップＳ２８）、処理を終了する。この復元処理では、合成処理により変換された文書を検索キーで検索し、あるレコードで検索キーと同じ部分文字列を内容に含む要素が検出されると、そのレコードにおける他の各要素の内容においても、検出された部分文字列の位置に対応する部分文字列を抽出する。そして、これらの部分文字列から、検索キーを含む変換前の複数個のレコードを復元し、検索結果として出力する。

【００６６】

次に、図１４は、図１３のステップＳ２３で行われる合成対象指定処理のフローチャートである。この処理では、兄弟要素の中で同じ名前の要素が複数個連続する場合、それらの要素を所定数の要素から成る複数の群に分割し、各群に含まれる要素に基づいて、合成対象が指定される。

【００６７】

変換装置は、まず、同名の兄弟要素をｎ個単位の群に分けるために、数値ｎを合成対象指定処理に入力する（ステップＳ３１）。次に、ＸＭＬ文書にＸＭＬプロセッサを適用して、木構造から成るＸＭＬ文書のオブジェクト（ＤＯＭ）を算

出し（ステップＳ３２）、要素数の集計処理を行う（ステップＳ３３～Ｓ３５）。

【００６８】

この処理では、まず、木構造のすべての要素に関する集計が終了したか否かをチェックする（ステップＳ３３）。集計が終了していなければ、木構造から各要素に属するＮｏｄｅＬｉｓｔ（連なるノードの一覧）を取得してメモリに格納する（ステップＳ３４）。そして、各ＮｏｄｅＬｉｓｔ内で兄弟の関係にある要素のノードにおいて、同じ要素名のものが連続して出現する回数を集計し（ステップＳ３５）、ステップＳ３３以降の処理を繰り返す。例えば、要素ａ、要素ａ、要素ｂが順に出現した場合、要素ａの連続出現回数は２回となる。

【００６９】

ステップＳ３３において、すべての要素に関する集計が終了すると、次に、木構造の兄弟の中で連続する同じ要素名の要素の各々について、その祖先または子孫のノードにおいて、やはり、兄弟の関係にある同じ要素名の連続があるか否かをチェックする（ステップＳ３６）。

【００７０】

そのような要素名の連続が検出されなければ、現在注目している層の兄弟の中でのみ同じ要素名の連続が存在することになるので、連続するそれらの要素を合成対象としてｎ個ずつの群に分割し、ｎ個の各ノードの位置を記録する（ステップＳ３７）。そして、得られたｎ個ずつの要素の群とそれらの位置情報を出力し（ステップＳ３８）、処理を終了する。

【００７１】

また、ステップＳ３６において、兄弟の関係にある同じ要素名の連続が祖先または子孫から検出されれば、注目している層を含む複数の層において同じ要素名の兄弟要素の連続が存在することになるので、そのような祖先または子孫を持つ要素を合成対象から外す。

【００７２】

次に、図１５は、図１３のステップＳ２４～Ｓ２７で行われる合成処理のフローチャートである。変換装置は、まず、合成対象指定処理から出力された、ｎ個

ずつの同名の兄弟要素の群とそれらの位置情報を、合成処理に入力し（ステップＳ４１）、入力されたすべての群についての合成が終了したか否かをチェックする（ステップＳ４２）。

【００７３】

合成が終了していない群があれば、次に、その群のすべての同名の兄弟要素ａについての処理が終了したか否かをチェックする（ステップＳ４３）。そして、処理が終了していない要素ａがあれば、その要素ａに連なる部分木を部分木ａとして、部分木ａの要素を探索する（ステップＳ４４～Ｓ５１）。同じ群に属する要素名ａの２つの要素を仮にａ１、ａ２と呼ぶことにすると、それらの要素に連なる部分木は、それぞれ部分木ａ１、部分木ａ２となる。

【００７４】

部分木ａの探索処理では、まず、部分木ａのルートである要素ａから下のすべての要素を探索したか否かをチェックする（ステップＳ４４）。探索していない任意の要素ｃがあれば、要素ａからその要素ｃに至るまでの経路を探索し、経路上の要素の名前の文字列を要素の流れとして記録する（ステップＳ４５）。

【００７５】

例えば、要素ａ、ｂ、ｃの順に上から辿った場合、（要素ａ）＋（要素ｂ）＋（要素ｃ）が要素の流れとして記録される。部分木ａ１に属する要素の流れ（要素ａ）＋（要素ｂ）＋（要素ｃ）と、部分木ａ２に属する要素の流れ（要素ａ）＋（要素ｂ）＋（要素ｃ）は、同じ要素の流れになる。

【００７６】

次に、記録された要素の流れの文字列を元に、ハッシュ関数の値（要素ｃのハッシュ値）を計算する（ステップＳ４６）。ここでは、例えば、要素名ａ、ｂ、ｃの文字列を文字符号に変換して得られる整数を、それぞれＩａ、Ｉｂ、Ｉｃとして、Ｉａ＊２５６、Ｉｂ＊１６、およびＩｃの排他的論理和ＥＸ（ａｂｃ）を求め、次式によりハッシュ値を計算する。

ハッシュ値＝ＥＸ（ａｂｃ）％ｍ

ただし、ｍは、２５６に対して素な整数であり、％ｍは、ｍによる剰余演算を表す。このとき、部分木ａ１と部分木ａ２にそれぞれ属する同じ要素の流れ同士は、互いに同じハッシュ値を持つことになる。

【００７７】

次に、現在の部分木における要素ｃが、同じ群の中で処理済みの一連の部分木ａにはなかった新しい要素か否かをチェックする（ステップＳ４７）。要素ｃが新しい要素であれば、ハッシュ表における要素ｃのハッシュ値の欄に、それまでに処理した部分木ａの数（要素ａの数）だけ、境界文字＠を並べて登録する（ステップＳ４８）。＠の数が多い場合は、連長表現を用いてもよい。

【００７８】

次に、同じ群の中で処理済みの一連の部分木ａとは異なり、現在の部分木において要素ｃが欠損しているか否かをチェックする（ステップＳ４９）。要素ｃが欠損していれば、ハッシュ表における要素ｃのハッシュ値の欄に、境界文字＠を追加して登録する（ステップＳ５０）。また、要素ｃが欠損していなければ、ハッシュ表における要素ｃのハッシュ値の欄に、要素ｃの内容と境界文字＠を追加して登録する（ステップＳ５１）。そして、ステップＳ４４以降の処理を繰り返す。

【００７９】

ステップＳ４６～Ｓ５２の処理により、異なる部分木に属する要素でも、要素の流れが等しいもの同士は、同じハッシュ値を持つことになり、それらの要素の内容は、同じハッシュ表の同じハッシュ値に対応する欄で、境界文字＠を用いて接合される。

【００８０】

ステップＳ４４において、すべての要素の探索が終了すると、ステップＳ４３以降の処理を繰り返す。そして、ステップＳ４３において、すべての同名の兄弟要素ａについての処理が終了すると、同じ群に属するｎ通りの部分木の接合が終了したことになる。そこで、次に、得られたハッシュ表の登録内容を元に、合成部分木を生成する（ステップＳ５２）。このとき、接合されなかった要素があれば、それより上位の要素から生成された合成部分木に含まれる、新しい要素の下

位に、その接合されなかった要素の複製を生成する。

【００８１】

次に、同じ群に属する処理済みのｎ通りの部分木を削除し、合成部分木のみを残して（ステップＳ５３）、ステップＳ４２以降の処理を繰り返す。そして、ステップＳ４２において、すべての群についての合成が終了すると、処理を終了する。このような合成処理によれば、複数の対応する要素内容が接合され、元の要素内容が削除されるので、文書が圧縮される。圧縮された文書は、外部記憶装置等に保存される。

【００８２】

次に、図１６は、圧縮されたＸＭＬ文書の上でタグ検索を行い、検索結果を出力する復元処理のフローチャートである。変換装置は、まず、圧縮されたＸＭＬ文書を復元処理に入力し（ステップＳ６１）、ユーザから指定された検索キーの文字列を復元処理に入力する（ステップＳ６２）。

【００８３】

次に、図２７と同様の処理により、圧縮されたＸＭＬ文書の上でタグ検索を行う（ステップＳ６３）。上述した合成処理では、元のＸＭＬ文書で同じ要素名ａを持つ連続した兄弟要素に連なる部分木を合成したが、基本的な要素の位置関係は変化していないため、透過性は保たれている。

【００８４】

合成部分木の要素内容の中で、検索キーと同じ部分文字列が検出された場合、その文字列の前に登録されている境界文字＠の個数ｄを計算する（ステップＳ６４）。このｄの値により、合成前の各部分木のうち、最初からｄ＋１番目の部分木に検索キーに相当する部分文字列があったことが分かる。

【００８５】

このとき、同じ要素内容の中の複数の個所で検索キーと同じ部分文字列が検出されれば、検出されたすべての文字列に対するｄの値を計算する。ｋ個の同じ部分文字列（０＜ｋ≦ｎ）が検出された場合、各々に対応する境界文字＠の個数としてd_1，．．．，d_kが計算される。

【００８６】

次に、同名の複数の要素ａに連なる合成部分木のうち、検索キーと同じ部分文字列が検出された要素内容を含む合成部分木を、メモリ上に格納する（ステップＳ６５）。以下の分離処理は、メモリ上に格納された合成部分木のみに対して行われるが、外部記憶装置に保存されている合成部分木は変更されない。

【００８７】

次に、メモリ上のすべての合成部分木について処理を行ったか否かをチェックする（ステップＳ６６）。未処理の合成部分木があれば、次に、未処理のある合成部分木のすべての要素を処理したか否かをチェックする（ステップＳ６７）。

【００８８】

未処理の要素があれば、その要素の内容において、d_i 番目（ｉ＝１，．．．，ｋ）の“@”とd_i ＋１番目の“@”に囲まれた部分文字列を残して、残りの文字列を削除する（ステップＳ６８）。そして、ステップＳ６７以降の処理を繰り返す。このような文字列の削除処理は、木構造全体の一部分のみに対して行われるので、削除処理に伴う処理速度の劣化は防止される。

【００８９】

これにより、すべての要素内容において、同じd_i の値の組み合わせに対応する個所の部分文字列が抽出され、１つの合成部分木を構成するｎ個の元の部分木のうち、検索キーを要素内容に含むものだけが再現される。

【００９０】

ステップＳ６７において、すべての要素の処理が終了すると、次に、再現された元の部分木をＸＭＬ文書に変換し（ステップＳ６９）、生成されたＸＭＬ文書を出力して（ステップＳ７０）、ステップＳ６６以降の処理を繰り返す。そして、ステップＳ６６において、メモリ上のすべての合成部分木の処理が終われば、復元処理を終了する。

【００９１】

ところで、欠損等がなく、完全に同じ部分構造が繰り返される文書の場合、ＤＯＭを生成する必要はなく、図１４の合成対象指定処理と図１５の合成処理の代わりに、上述した簡易型合成処理を用いることができる。

【００９２】

図１７は、このような簡易型合成処理のフローチャートである。変換装置は、まず、数値ｎを簡易型合成処理に入力し（ステップＳ７１）、図４のＥｖｅｎｔ要素のように、同じ層で連続する要素名ａを入力する（ステップＳ７２）。そして、要素名ａの出現頻度を表す変数Ｆａを初期化してＦａ＝０とおき（ステップＳ７３）、登録に用いるハッシュ表の内容を初期化する（ステップＳ７４）。

【００９３】

次に、ＸＭＬ文書内のすべてのデータの走査が終了したか否かをチェックする（ステップＳ７５）。走査されていない部分があれば、その部分から連続して存在する要素名ａを検出し（ステップＳ７６）、Ｆａに１を加算する（ステップＳ７７）。

【００９４】

次に、要素ａに連なる各要素ｂを検出し（ステップＳ７８）、ハッシュ表の要素ｂのハッシュ値に対応する欄に、要素ｂの内容と境界文字＠を追加登録する（ステップＳ７９）。ここでは、要素ｂの親は要素ａであることが分かっているため、要素ｂのハッシュ値の計算では、要素ａの文字列を省略することができる。そこで、例えば、要素ｂの文字列を文字符号に変換して得られる整数をＩｂとして、次式によりハッシュ値が計算される。

ハッシュ値＝Ｉｂ％ｍ

次に、Ｆａがｎの倍数になったか否かをチェックし（ステップＳ８０）、Ｆａがｎの倍数でなければ、ステップＳ７５以降の処理を繰り返す。これにより、複数の要素ａに連なる同じ層の要素の内容同士が接合される。

【００９５】

ステップＳ８０において、Ｆａがｎの倍数であれば、１つの群の合成処理が終了したことになるので、ハッシュ表に登録された、接合された要素内容を元に、部分的な文書構造を生成し、出力する（ステップＳ８１）。次に、次の群の合成処理に備えて、ハッシュ表で要素ｂのために使用された欄を初期化し（ステップＳ８２）、ステップＳ７５以降の処理を繰り返す。

【００９６】

そして、ステップＳ７５において、すべてのデータの走査が終了すると、処理を終了する。こうして生成されたＸＭＬ文書の上でタグ検索を行う場合は、やはり、ＤＯＭを利用するの有効であるので、上述した図１６の復元処理が用いられる。

【００９７】

以上説明した実施形態においては、主として、ＸＭＬ文書のタグ検索を例に用いているが、本発明は、ＸＭＬ文書以外の構造化文書に対しても適用可能である。

【００９８】

本実施形態の変換装置は、例えば、図１８に示すような情報処理装置（コンピュータ）を用いて構成される。図１８の情報処理装置は、ＣＰＵ（中央処理装置）８１、メモリ８２、入力装置８３、出力装置８４、外部記憶装置８５、媒体駆動装置８６、およびネットワーク接続装置８７を備え、それらはバス８８により互いに接続されている。

【００９９】

メモリ８２は、例えば、ＲＯＭ（read only memory）、ＲＡＭ（random access memory）等を含み、処理に用いられるプログラムとデータを格納する。ＣＰＵ８１は、メモリ８２を利用してプログラムを実行することにより、必要な処理を行う。

【０１００】

入力装置８３は、例えば、キーボード、ポインティングデバイス、タッチパネル等であり、ユーザからの指示や情報の入力に用いられる。出力装置８４は、例えば、ディスプレイ、プリンタ、スピーカ等であり、ユーザへの問い合わせや処理結果の出力に用いられる。

【０１０１】

外部記憶装置８５は、例えば、磁気ディスク装置、光ディスク装置、光磁気ディスク（magneto-optical disk）装置、テープ装置等である。情報処理装置は、この外部記憶装置８５に、上述のプログラムとデータを保存しておき、必要に応

じて、それらをメモリ８２にロードして使用する。

【０１０２】

媒体駆動装置８６は、可搬記録媒体８９を駆動し、その記録内容にアクセスする。可搬記録媒体８９としては、メモリカード、フロッピーディスク、ＣＤ－ＲＯＭ（compact disk read only memory）、光ディスク、光磁気ディスク等、任意のコンピュータ読み取り可能な記録媒体が用いられる。ユーザは、この可搬記録媒体８９に上述のプログラムとデータを格納しておき、必要に応じて、それらをメモリ８２にロードして使用する。

【０１０３】

ネットワーク接続装置８７は、ＬＡＮ（local area network）等の任意の通信ネットワークに接続され、通信に伴うデータ変換を行う。また、情報処理装置は、上述のプログラムとデータをネットワーク接続装置８７を介して、サーバ等の他の装置から受け取り、必要に応じて、それらをメモリ８２にロードして使用する。

【０１０４】

図１９は、図１８の情報処理装置にプログラムとデータを供給することのできるコンピュータ読み取り可能な記録媒体を示している。可搬記録媒体８９やサーバ９０のデータベース９１に保存されたプログラムとデータは、メモリ８２にロードされる。そして、ＣＰＵ８１は、そのデータを用いてそのプログラムを実行し、必要な処理を行う。このとき、サーバ９０は、プログラムとデータを伝送する伝搬信号を生成し、ネットワーク上の任意の伝送媒体を介して、情報処理装置に送信する。

（付記１）　階層構造の要素の集合で記述され、それぞれが１つ以上の要素を含む複数のレコードから成る構造化文書の情報を入力する文書入力手段と、

前記構造化文書の２つ以上のレコード間で、相対的に同じ位置にある要素の内容を接合して、新しい要素を生成する接合手段と、

前記新しい要素を含み、前記２つ以上のレコードにおける要素の相対的位置関係を継承した新しいレコードを生成する生成手段と、

前記２つ以上のレコードを前記新しいレコードに置き換えて、前記構造化文書

を変換する変換手段と、

変換後の構造化文書を出力する文書出力手段と

を備えることを特徴とする変換装置。

（付記２）　検索キーを入力するキー入力手段と、前記変換後の構造化文書を該検索キーで検索し、あるレコードの要素の内容から前記検索キーに対応する文字列が検出されたとき、該あるレコードにおける他の要素の内容から、検出された文字列の位置に対応する文字列を抽出し、該検出された文字列と抽出された文字列から該検索キーを含む変換前のレコードを復元し、検索結果として出力する検索手段とをさらに備えることを特徴とする付記１記載の変換装置。

（付記３）　階層構造の要素の集合で記述された構造化文書の情報を入力する文書入力手段と、

前記構造化文書の情報を格納する格納手段と、

前記構造化文書において、ある要素の１段下の層で連続して並ぶ同じ要素名の要素同士の組み合わせと、該組み合わせの各要素より下位のある層の同じ要素名の要素同士であって、該組み合わせの各要素から該ある層に至る経路上の各層において互いに同じ要素名の要素を経由するような、該ある層の要素同士の組み合わせとに含まれる各要素の内容を合成対象として接合し、複数の新しい要素を生成する接合手段と、

前記複数の新しい要素を含み、該複数の新しい要素の間で元の要素の相対的位置関係を継承した合成部分構造を生成する生成手段と、

接合されなかった要素より上位の要素から生成された合成部分構造に含まれる新しい要素の下位に、該接合されなかった要素の複製を生成する複製手段と、

不要な元の要素を削除する削除手段と、

前記接合手段、生成手段、複製手段、および削除手段を用いて、前記構造化文書を合成部分構造から成る合成型構造化文書に変換する変換手段と、

前記合成型構造化文書を出力する文書出力手段と

を備えることを特徴とする変換装置。

（付記４）　前記生成手段は、前記ある層に至る経路上の２つ以上の層において、連続して並ぶ同じ要素名の要素同士の組み合わせが見られないとき、前記合成

部分構造を生成することを特徴とする付記３記載の変換装置。

（付記５）　前記接合手段は、前記ある要素の１段下の層の要素同士の組み合わせを、所定数の要素から成る複数の群に分割し、各群に含まれる該所定数の要素に基づいて、前記合成対象を指定することを特徴とする付記３記載の変換装置。

（付記６）　前記接合手段は、接合される２つの内容の間に境界文字を挿入して、前記新しい要素の内容を生成することを特徴とする付記３記載の変換装置。

（付記７）　前記接合手段は、前記合成対象となる要素の内容が欠損しているとき、前記新しい要素の内容に前記境界文字を連続して挿入することを特徴とする付記６記載の変換装置。

（付記８）　検索キーを入力するキー入力手段と、前記合成型構造化文書内の要素の内容に含まれる２つの境界文字の間の文字列と該検索キーの文字列とを照合し、ある合成部分構造の要素の内容から該検索キーに対応する文字列が検出されたとき、検出された文字列の前にある境界文字の順位を求め、該ある合成部分構造における他の要素の内容において、該順位に対応する境界文字と次の境界文字の間の文字列を抽出し、該検出された文字列と抽出された文字列から変換前の構造化文書の対応する部分を復元し、検索結果として出力する検索手段とをさらに備えることを特徴とする付記６記載の変換装置。

（付記９）　コンピュータのためのプログラムを記録した記録媒体であって、該プログラムは、

階層構造の要素の集合で記述され、それぞれが１つ以上の要素を含む複数のレコードから成る構造化文書の２つ以上のレコード間で、相対的に同じ位置にある要素の内容を接合して、新しい要素を生成し、

前記新しい要素を含み、前記２つ以上のレコードにおける要素の相対的位置関係を継承した新しいレコードを生成し、

前記２つ以上のレコードを前記新しいレコードに置き換えて、前記構造化文書を変換する

処理を前記コンピュータに実行させることを特徴とするコンピュータ読み取り可能な記録媒体。

（付記１０）　コンピュータにプログラムを伝送する伝搬信号であって、該プロ

グラムは、

階層構造の要素の集合で記述され、それぞれが１つ以上の要素を含む複数のレコードから成る構造化文書の２つ以上のレコード間で、相対的に同じ位置にある要素の内容を接合して、新しい要素を生成し、

前記新しい要素を含み、前記２つ以上のレコードにおける要素の相対的位置関係を継承した新しいレコードを生成し、

前記２つ以上のレコードを前記新しいレコードに置き換えて、前記構造化文書を変換する

処理を前記コンピュータに実行させることを特徴とする伝搬信号。

【０１０５】

【発明の効果】

本発明によれば、構造化文書の要素の数が減り、文書が圧縮されるため、構造化文書を格納するためのメモリ量が削減される。部分木の合成を行っても、要素間の基本的な関係は維持されるため、従来の応用ソフトウェアの処理に対する変換による影響はなく、処理の透過性が保たれる。

【０１０６】

また、木構造においてノードが減少するために、探索に要する処理時間が大幅に削減され、タグ検索の処理速度が向上する。例えば、オフラインで、元の構造化文書から合成部分木をあらかじめ生成しておけば、随時、その合成部分木をタグ検索に用いることができ、変換処理の時間はタグ検索の時間には含まれない。

【図面の簡単な説明】

【図１】

本発明の変換装置の原理図である。

【図２】

第１の処理対象のＸＭＬ文書を示す図である。

【図３】

第１の処理対象の木構造を示す図である。

【図４】

第２の処理対象のＸＭＬ文書を示す図である。

【図５】

第２の処理対象の木構造を示す図である。

【図６】

合成後の木構造を示す図である。

【図７】

第１の合成後のＸＭＬ文書を示す図である。

【図８】

第１のハッシュ表を示す図である。

【図９】

第２のハッシュ表を示す図である。

【図１０】

第３の処理対象のＸＭＬ文書を示す図である。

【図１１】

第２の合成後のＸＭＬ文書を示す図である。

【図１２】

検索結果を示す図である。

【図１３】

ＸＭＬ文書の変換を含むタグ検索処理のフローチャートである。

【図１４】

合成対象指定処理のフローチャートである。

【図１５】

要素合成処理のフローチャートである。

【図１６】

復元処理のフローチャートである。

【図１７】

簡易型合成処理のフローチャートである。

【図１８】

情報処理装置の構成図である。

【図１９】

記録媒体を示す図である。

【図２０】

ＸＭＬ文書の構成部分を示す図である。

【図２１】

タグの書き方を示す図である。

【図２２】

要素の階層構造を示す図である。

【図２３】

構造化文書の構成と処理上の区分との関係を示す図である。

【図２４】

ＸＭＬプロセッサの処理を示す図である。

【図２５】

ＸＭＬ文書を示す図である。

【図２６】

ＤＯＭの木構造を示す図である。

【図２７】

ＤＯＭを用いたタグ検索処理のフローチャートである。

【図２８】

ＳＡＸを用いたタグ検索処理のフローチャートである。

【符号の説明】

１１　ＸＭＬ宣言

１２　文書型定義

１３　ＸＭＬ実現値

２１　＜要素名＞

２２　＜／要素名＞

２３　＜要素名／＞

３１　ＸＭＬ文書

３２　ＸＭＬプロセッサ

３３　木構造

３４　応用ソフトウェア

４１　Ｄｏｃｕｍｅｎｔ

４２　ＮｏｄｅＬｉｓｔ

４３　Ｅｌｅｍｅｎｔ

４４、７５　Ｔｅｘｔ

４５　ＮａｍｅｄＮｏｄｅＭａｐ

４６　Ａｔｔｒ

５１　文書入力手段

５２　接合手段

５３　生成手段

５４　変換手段

５５　文書出力手段

５６　格納手段

５７　複製手段

５８　削除手段

６１、７１　Ｅｖｅｎｔｌｉｓｔ

６２、７２　Ｅｖｅｎｔ

６３、７４　Ｉｎｆｏ

７３　Ｓｔａｒｔ

８１　ＣＰＵ

８２　メモリ

８５　外部記憶装置

８６　媒体駆動装置

８７　ネットワーク接続装置

８８　バス

８９　可搬記録媒体

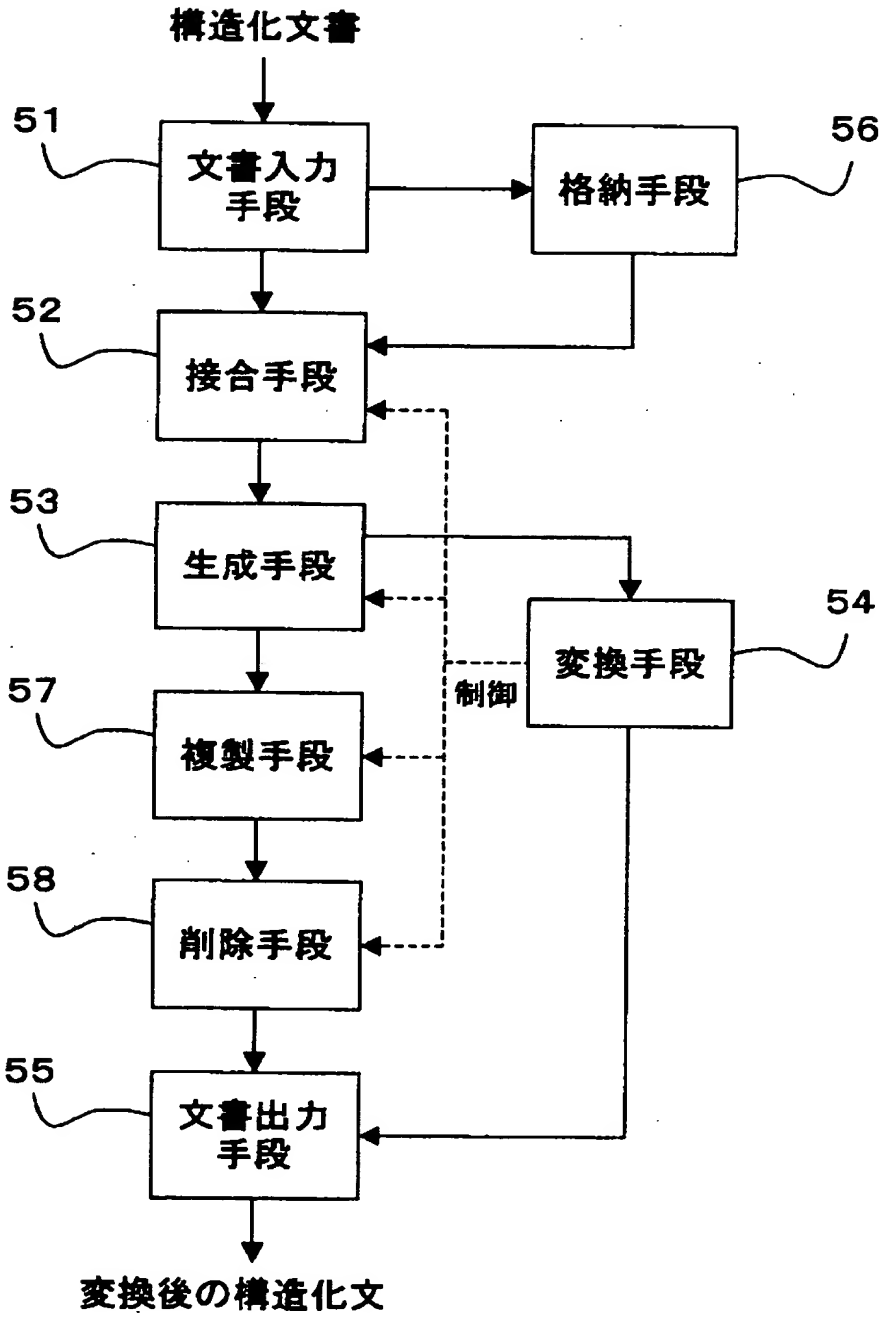
９０　サーバ

９１　データベース

【書類名】　　図面

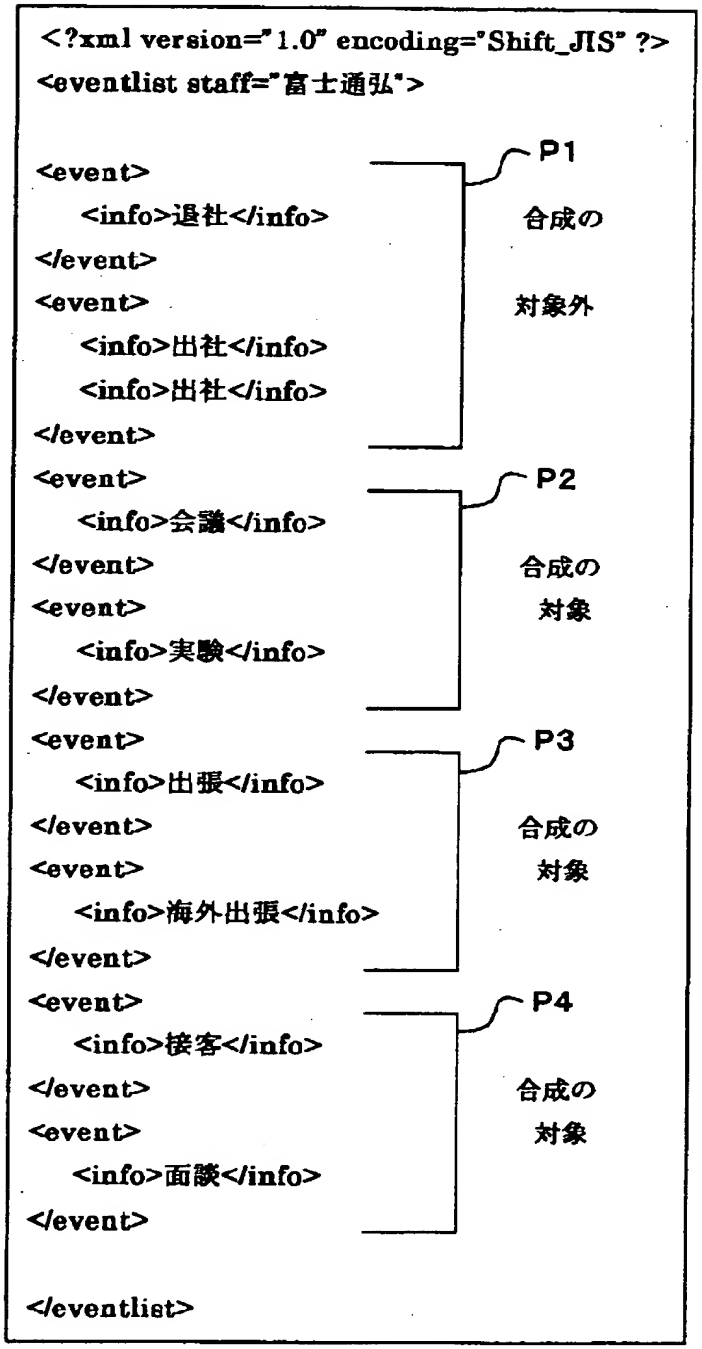
【図１】

本発明の原理図



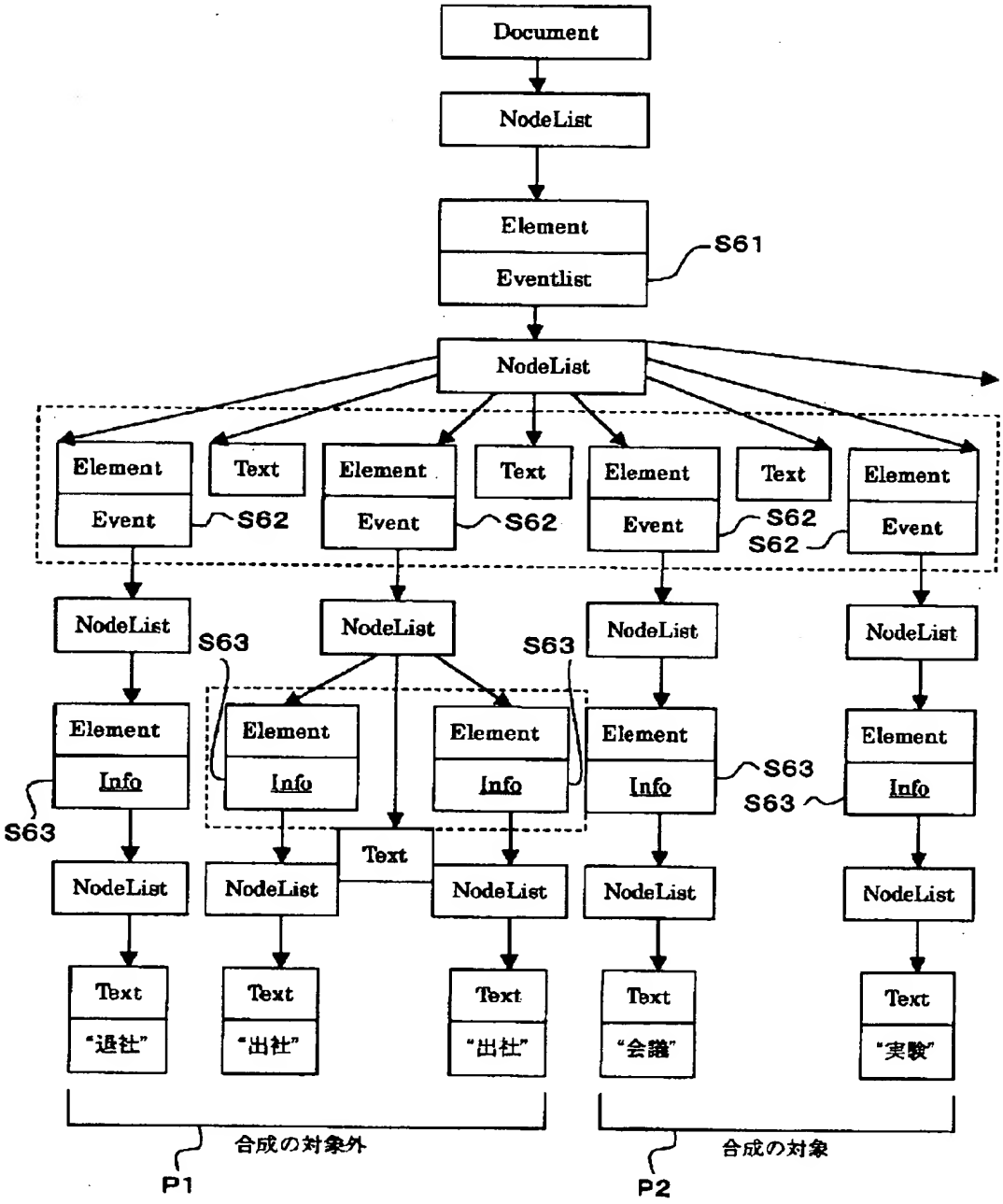
【図２】

第1の処理対象のXML文書を示す図



【図３】

第1の処理対象の木構造を示す図



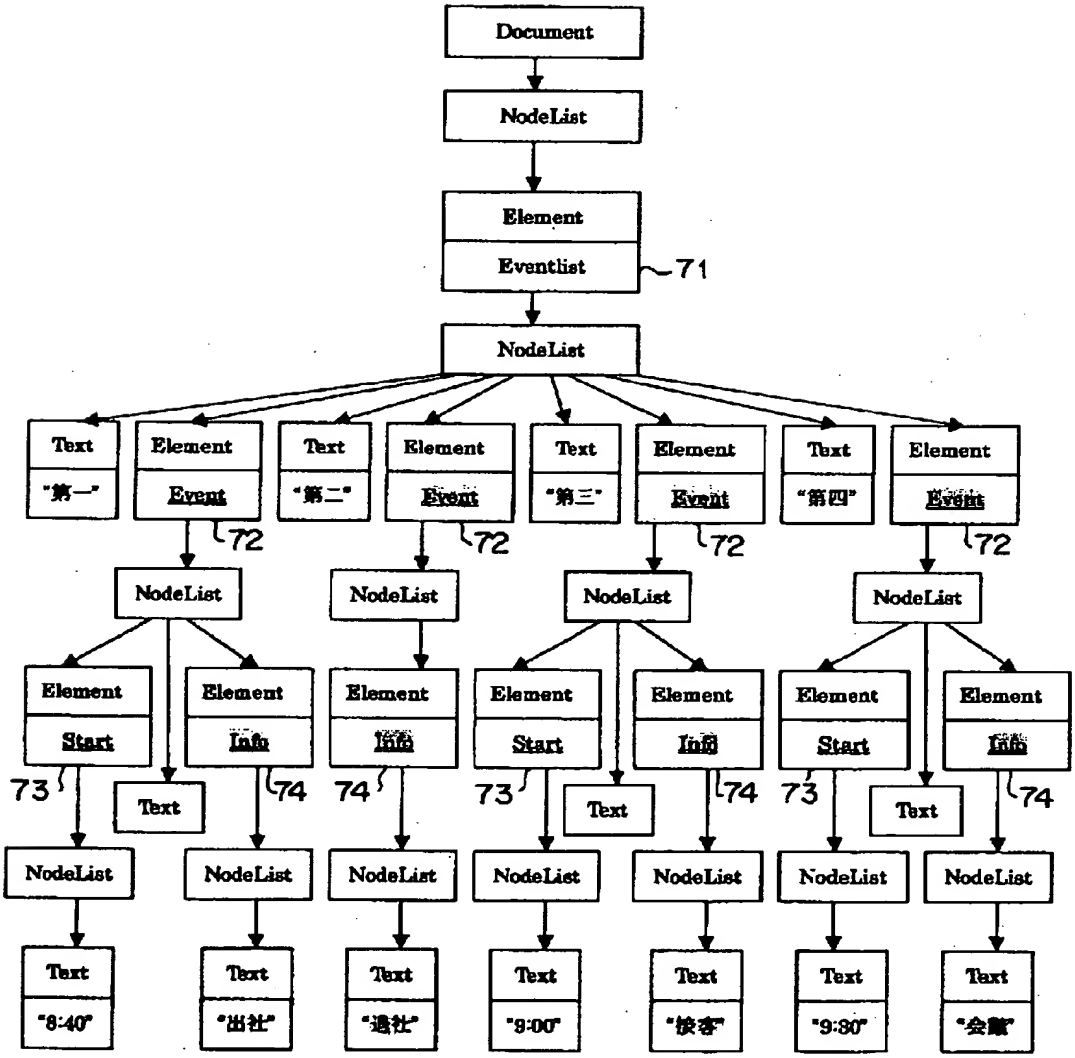
【図４】

第2の処理対象のXML文書を示す図

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<eventlist staff="富士通弘">
        第一
        <event>
                <start>8:40</start>
                <info>出社</info>
        </event>
        第二
        <event>
                <info>退社</info>
        </event>
        第三
        <event>
                <start>9:00</start>
                <info>接客</info>
        </event>
        第四
        <event>
                <start>9:30</start>
                <info>会議</info>
        </event>
</eventlist>
```

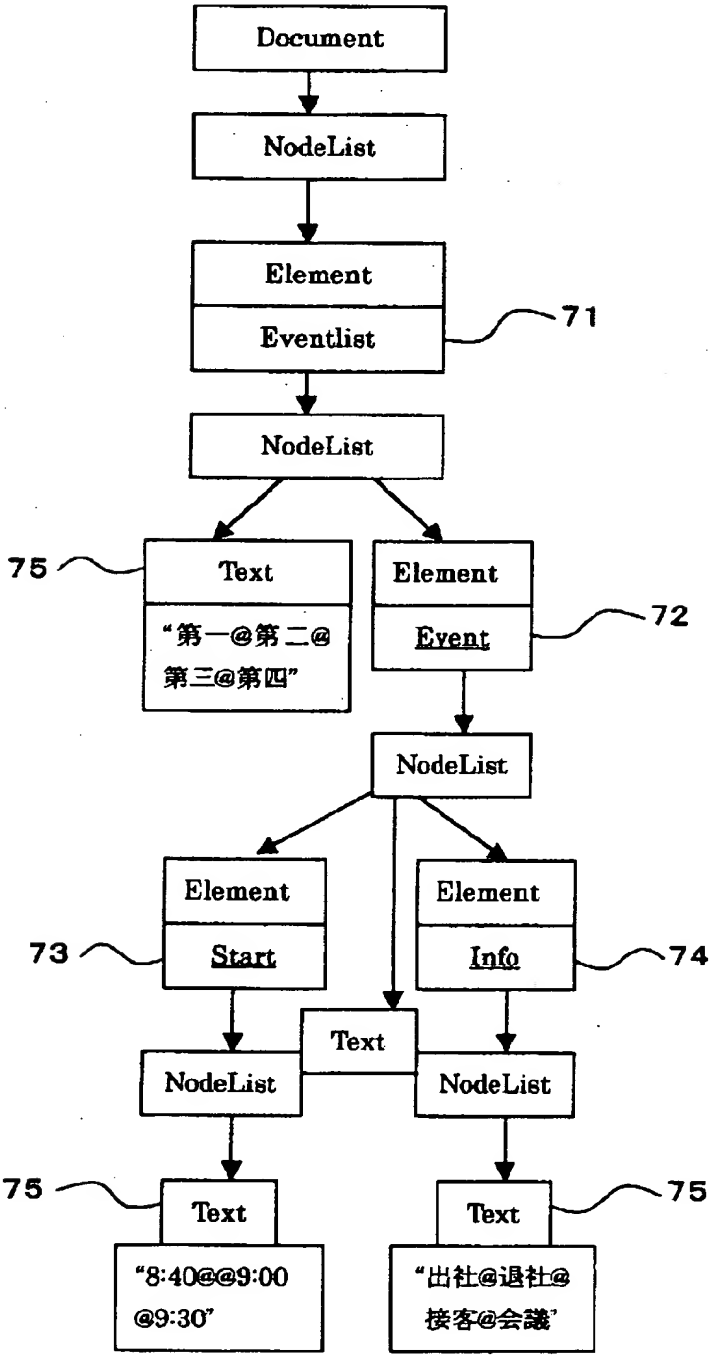
【図５】

第2の処理対象の木構造を示す図



【図６】

合成後の木構造を示す図



【図７】

第1の合成後のXML文書を示す図

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<eventlist staff="富士通弘">
        第一@第二@第三@第四
        <event>
                <start>8:40@@9:00@9:30</start>
                <info>出社@退社@接客@会議</info>
        </event>
</eventlist>
```

【図８】

第1のハッシュ表を示す図

ハッシュ値	先祖、親の要素名	自身の要素名	要素内容
H 1	Event	Info	出社@退社@接客@会議
H 2	Event	Start	8:40@@9:00@9:30

【図９】

第2のハッシュ表を示す図

ハッシュ値	先祖、親の要素名	自身の要素名	要素内容
H 3	Event	Info	出社@退社@接客@会議
H 4	Event	Start	8:40@17:00@9:00@9:30

【図１０】

第3の処理対象のXML文書を示す図

```
<?XML version="1.0"?>
<DOCTYPE doc>
<doc>
<個人>
<姓>鈴木</姓>
<名>小百合</名>
<旧姓>小川</旧姓>
</個人>
<個人>
<姓>佐藤</姓>
<名>一郎</名>
</個人>
</doc>
```

【図１１】

第2の合成後のXML文書を示す図

```
<?XML version="1.0"?>
<DOCTYPE doc>
<doc>
<個人>
<姓>鈴木@佐藤</姓>
<名>小百合@一郎</名>
<旧姓>小川@</旧姓>
</個人>
</doc>
```

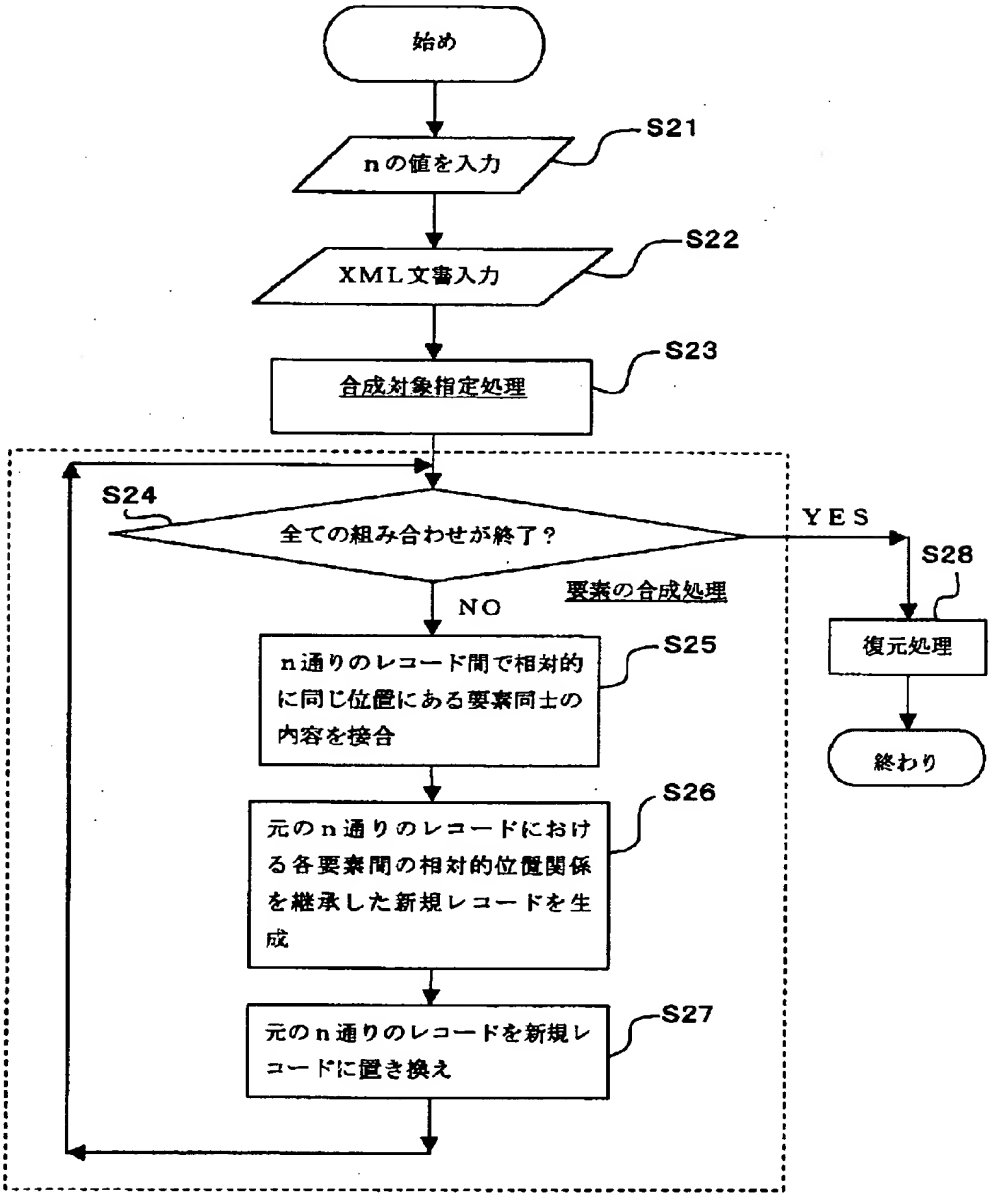
【図１２】

検索結果を示す図

```
<?XML version="1.0"?>
<DOCTYPE doc>
<doc>
<個人>
<姓>鈴木</姓>
<名>小百合</名>
<旧姓>小川</旧姓>
</個人>
</doc>
```

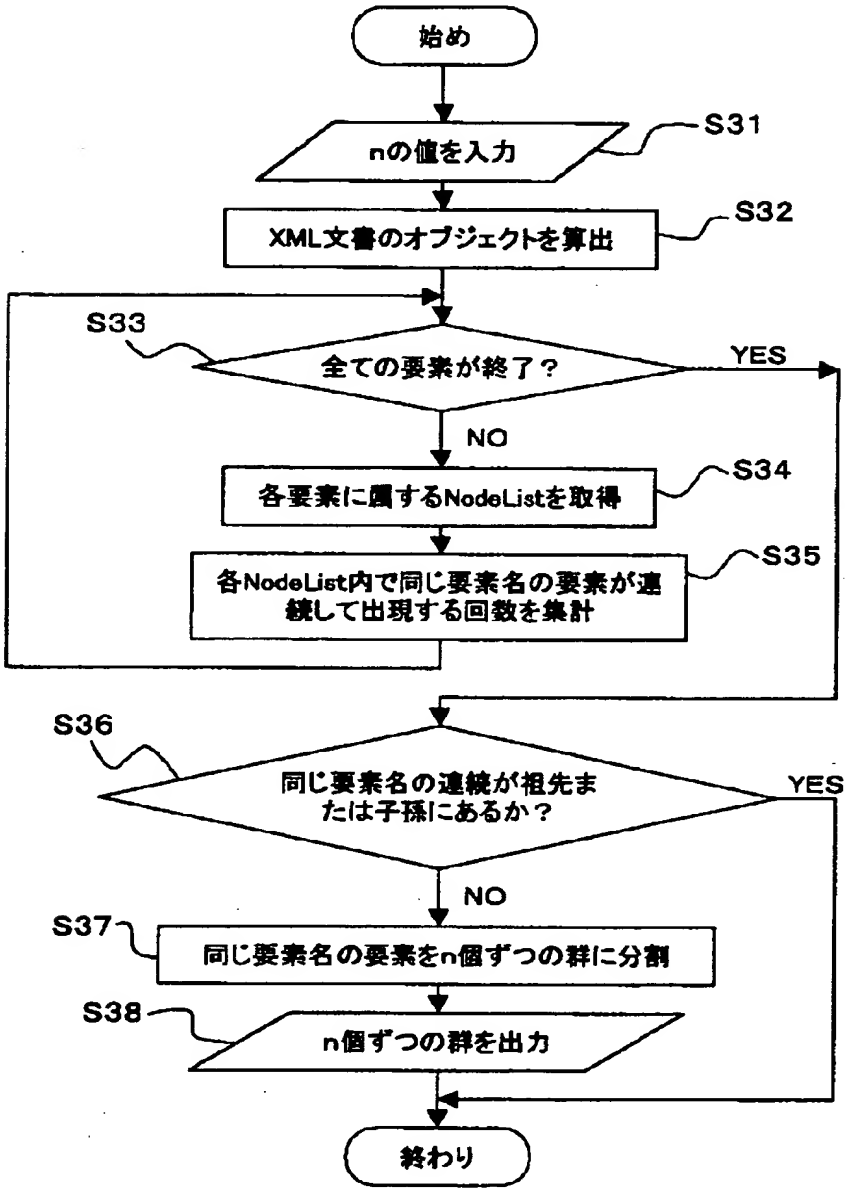
【図１３】

XML文書の変換を含むタグ検索処理のフローチャート



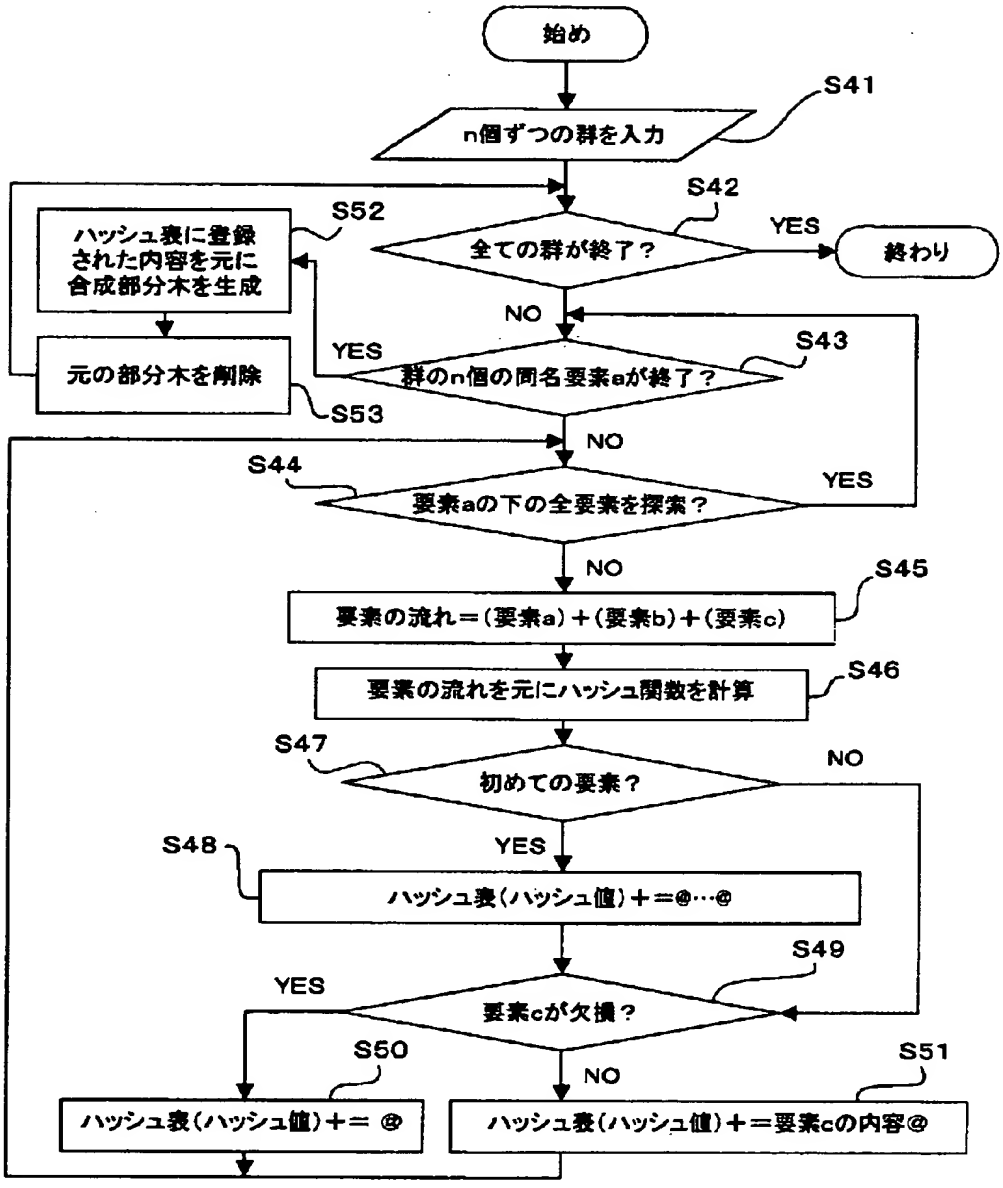
【図１４】

合成対象指定処理のフローチャート



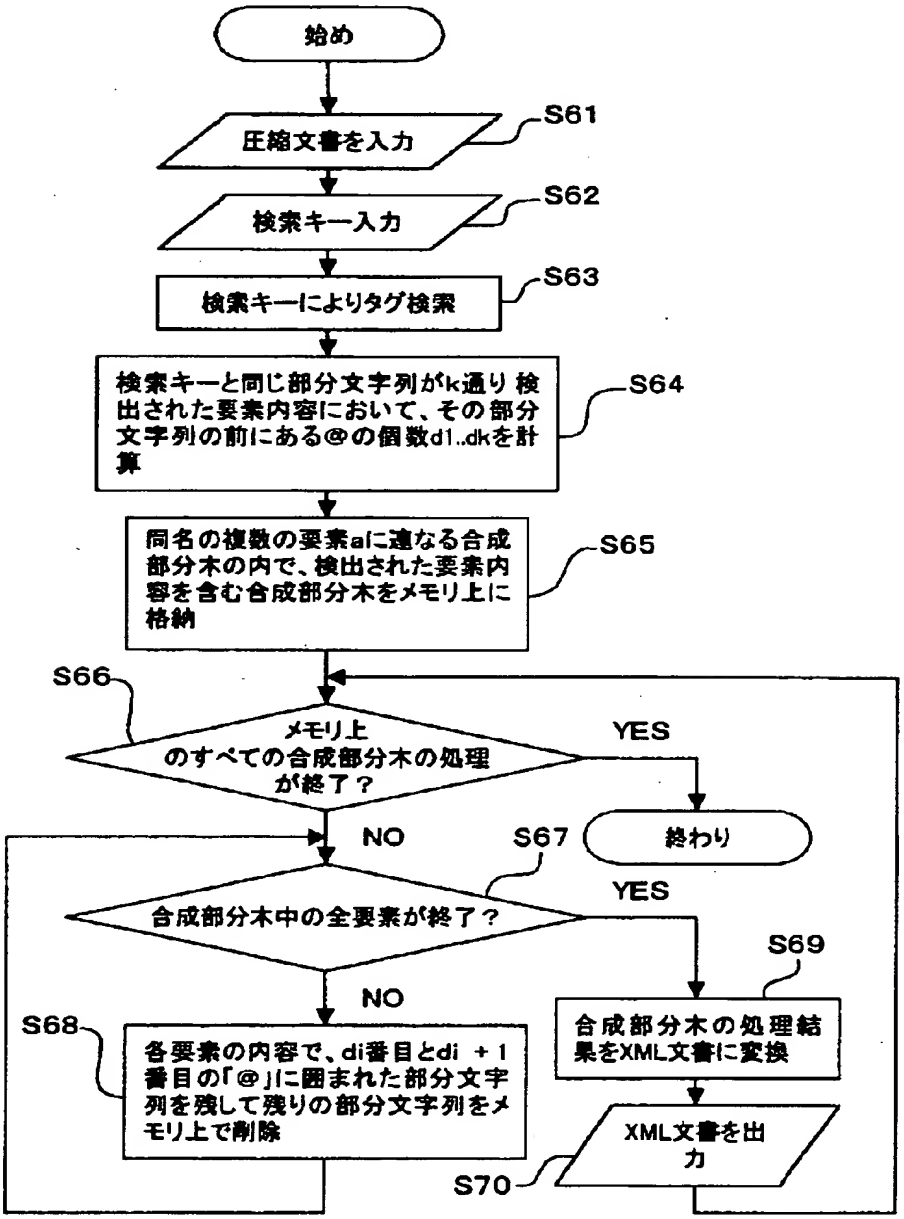
【図１５】

要素合成処理のフローチャート



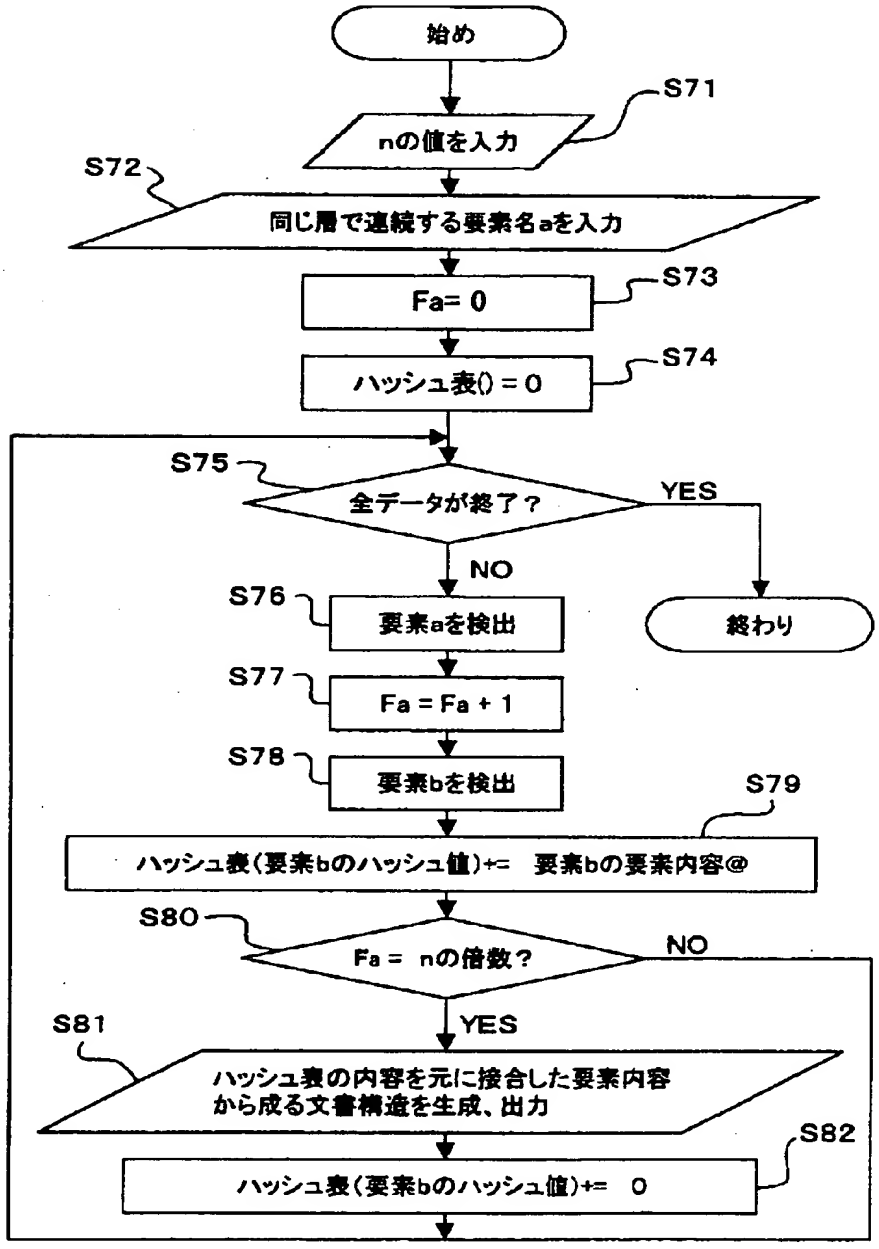
【図１６】

復元処理のフローチャート



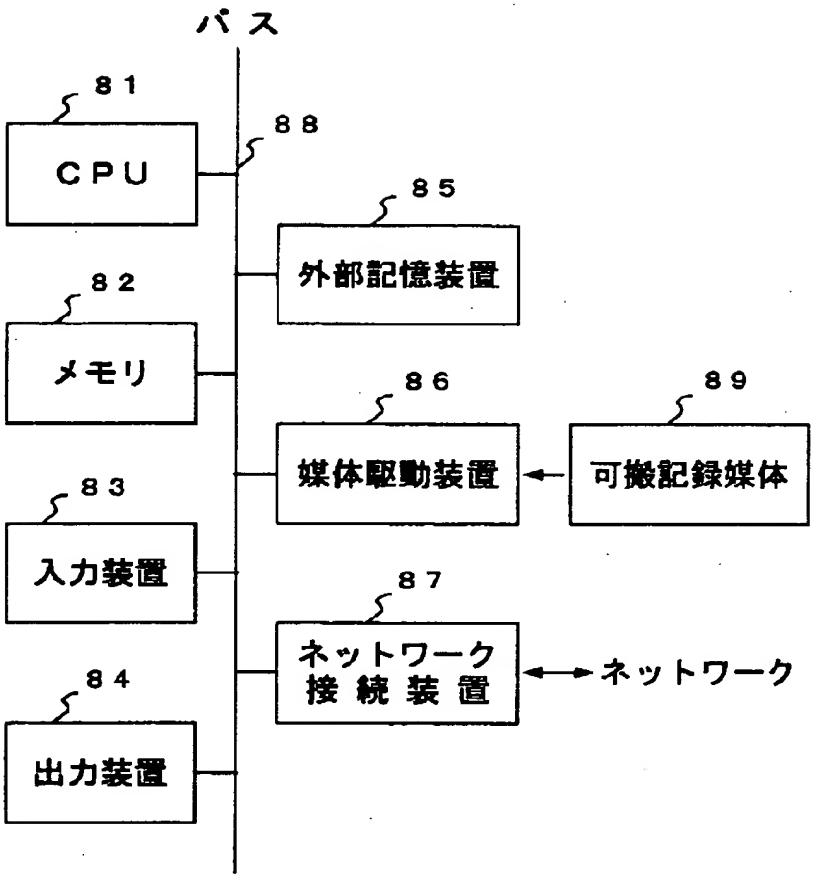
【図１７】

簡易型合成処理のフローチャート



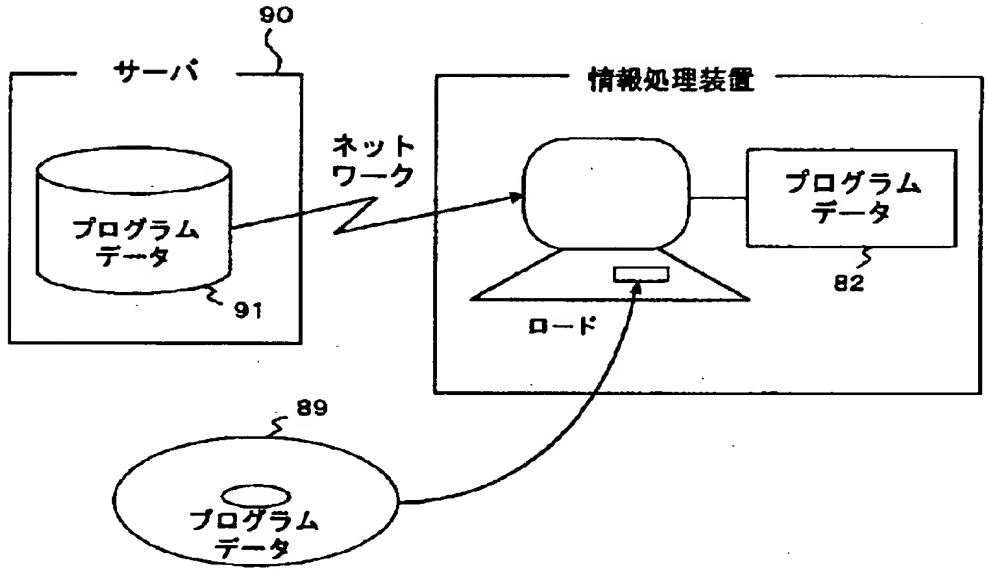
【図１８】

情　報　処　理　装　置　の　構　成　図



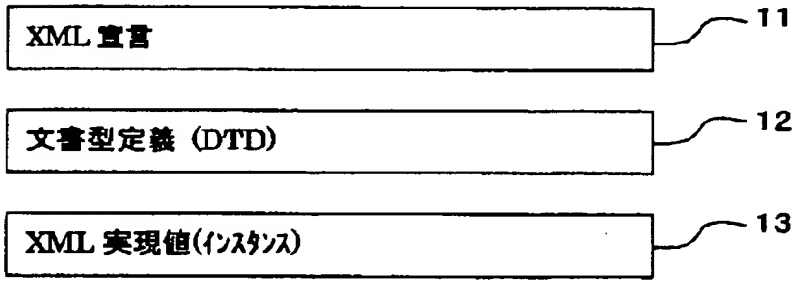
【図１９】

記　録　媒　体　を　示　す　図



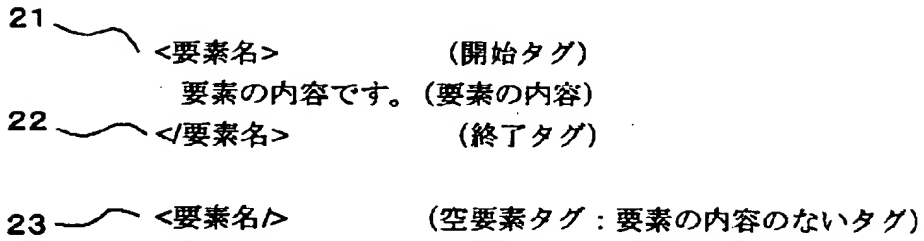
【図２０】

XML文書の構成部分を示す図



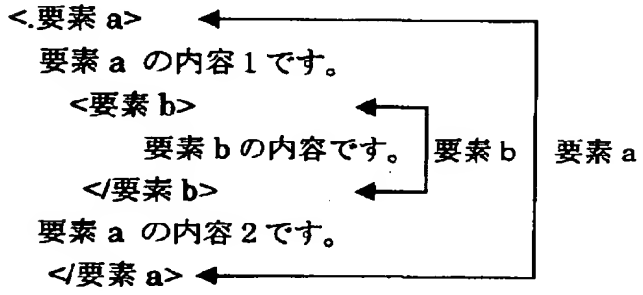
【図２１】

タグの書き方を示す図



【図２２】

要素の階層構造を示す図



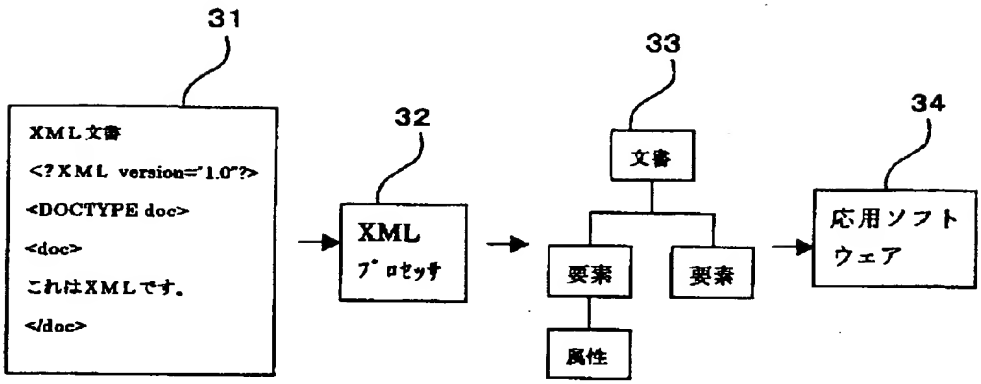
【図２３】

構造化文書の構成と処理上の区分との関係を示す図

	各種宣言	文書型定義	実現値
整形式 XML 文書	△ (XML 宣言)	△	○
検証済み XML 文書	△(XML 宣言)	○	○
SGML 文書	○(SGML 宣言)	○	○
HTML 文書	△(HTML 宣言)	○	○

【図２４】

XMLプロセッサの処理を示す図



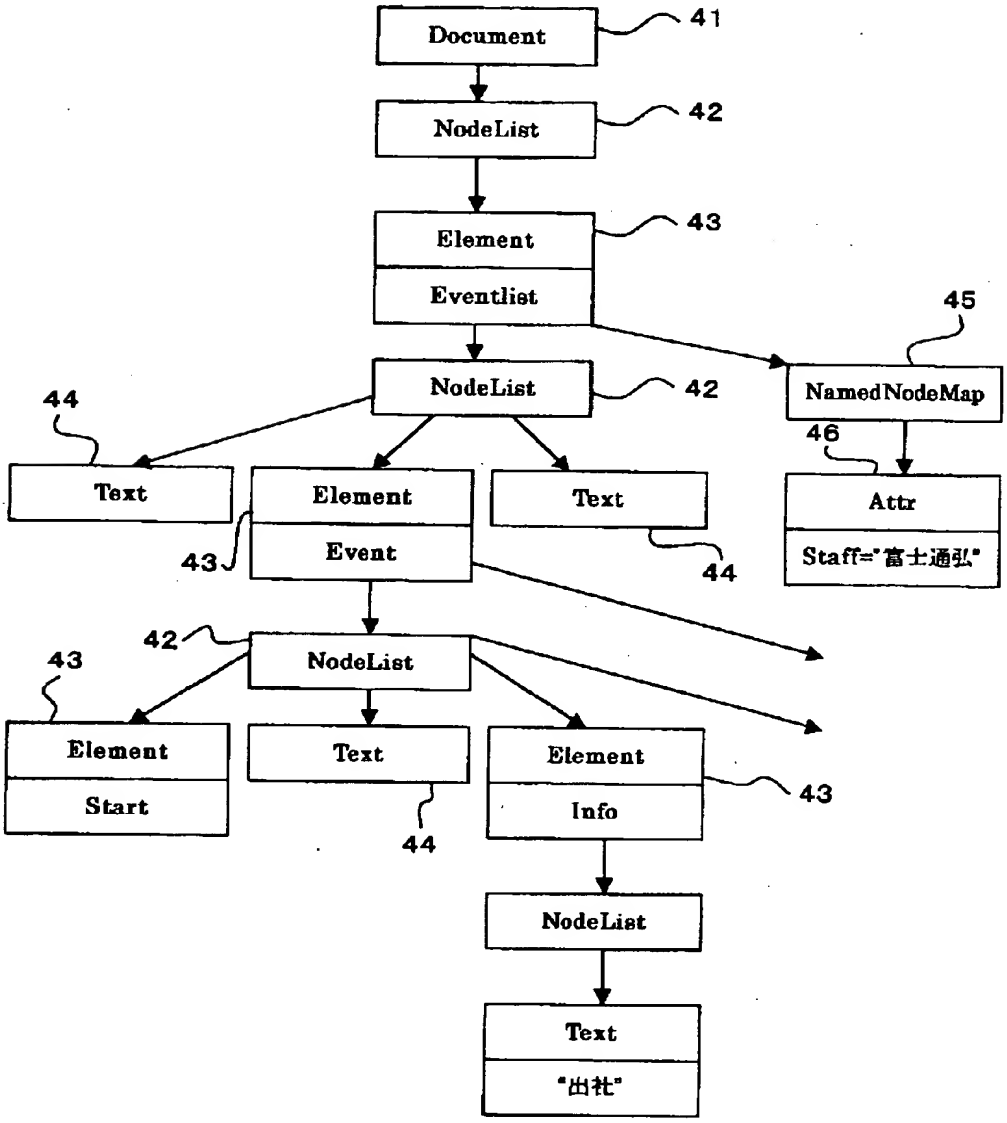
【図２５】

XML文書を示す図

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<eventlist staff="富士通弘">
  <event type="業務">
    <start/>
    <info>出社</info>
  </event>
</eventlist>
```

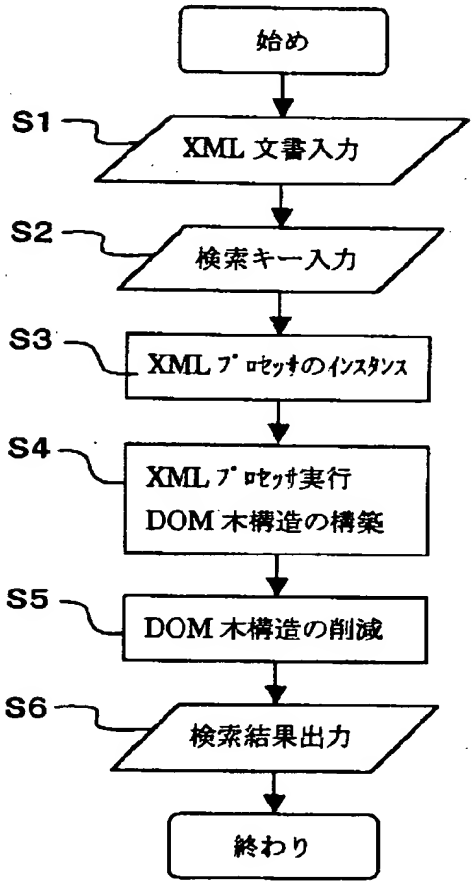
【図２６】

DOMの木構造を示す図



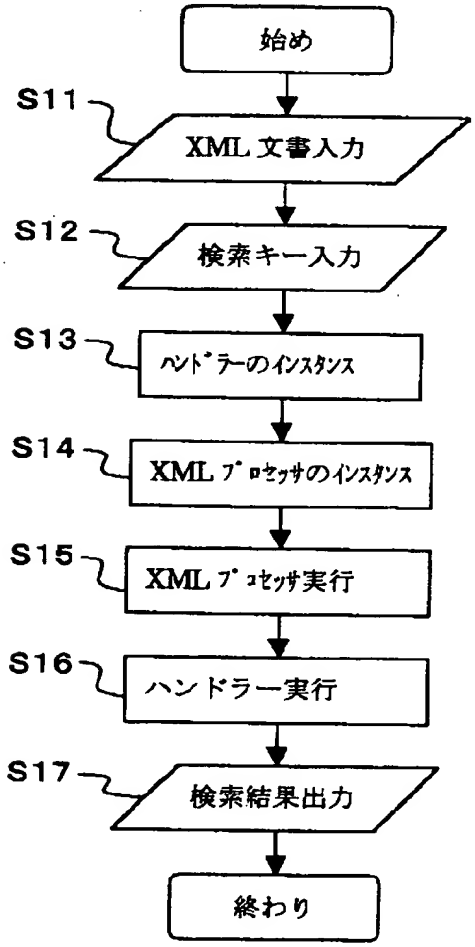
【図２７】

DOMを用いたタグ検索処理のフローチャート



【図２８】

SAXを用いたタグ検索処理のフローチャート



【書類名】　　要約書

【要約】

【課題】　ＸＭＬ文書のような構造化文書を変換することにより、タグ検索の処理速度を向上させ、必要な動作メモリ量を削減することが課題である。

【解決手段】　構造化文書を構成するＰ２、Ｐ３、およびＰ４の各部分に含まれる複数のレコードを合成対象として、それらのレコード間で同じ位置にある要素の内容を接合し、接合された内容を持つ新しい要素を生成する。そして、新しい要素を用いて、元のレコードにおける要素の相対的位置関係を継承した新しいレコードを生成し、元のレコードを新しいレコードに置き換えて、文書を変換する。

【選択図】　　図２

出　願　人　履　歴　情　報

識別番号　　〔０００００５２２３〕

１．変更年月日　　１９９６年　３月２６日
［変更理由］　住所変更
住　所　　神奈川県川崎市中原区上小田中４丁目１番１号
氏　名　　富士通株式会社